



[Keldysh Institute](#) • [Publication search](#)

[Keldysh Institute preprints](#) • [Preprint No. 73, 2020](#)



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

[J.C. Molina Saqui](#), [S.S. Tkachev](#)

Testbench calibration
technique for testing attitude
determination algorithms by
video processing

Recommended form of bibliographic references: Molina Saqui J.C., Tkachev S.S. Testbench calibration technique for testing attitude determination algorithms by video processing // Keldysh Institute Preprints. 2020. No. 73. 32 p. <https://doi.org/10.20948/prepr-2020-73-e>
<https://library.keldysh.ru/preprint.asp?id=2020-73&lg=e>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

J. C. Molina Saqui, S. Tkachev

**Testbench calibration technique
for testing attitude determination
algorithms by video processing**

Москва — 2020

Молина Саки Х. С., Ткачев С.С.

Методика калибровки стенда для отработки алгоритмов определения ориентации по видеоизображению

Работа посвящена задаче определения ориентации объекта с помощью обработки изображений. Для ее решения реализован двухэтапный подход. На первом этапе определяется матрица поворота через модель измерения, которая адаптирована для использования кватернионов. На втором этапе выполняется коррекция кватерниона ориентации с помощью метода наименьших квадратов.

Результаты экспериментальных исследований показали, что использование модели измерения камеры и итеративного процесса позволяет определить кватернион ориентации с хорошей точностью.

Ключевые слова: определение углового движения, обработка видеоизображения

Molina Saqui J. C., Tkachev S.

Testbench calibration technique for testing attitude determination algorithms by video processing

The paper is devoted to the problem of estimating the orientation of an object using image processing. A two-stage approach has been implemented to solve this problem. At the first stage the rotation matrix is determined using a measurement model adapted for using quaternions. At the second stage the orientation quaternion is corrected using the least squares method.

The results of experimental studies have shown that using a camera measurement model and an iterative process it is possible to determine the orientation quaternion with good accuracy.

Key words: angular motion determination, image processing

Introduction

Determination of angular motion is of great importance because it allows to know and predict the attitude of the bodies with respect to a reference system, such information is vital for missions where maneuvers and interactions of two or more bodies are performed.

Studies on objects attitude and angular velocity estimation have been performed using different sensors such as photoelectric encoders, tachometers, inertial sensors, and even laser. However, their implementation can be expensive. The use of digital images as low-cost sources of information for evaluating the angular motion is actively used in the field of robotics, control system, augmented reality, and are also widely used in the field of satellite systems.

Over the last four decades, a variety studies have been done on measuring motion parameters of objects using cameras, where a considerable importance had the development of methods for camera calibration, i.e. the determination of internal parameters of the camera. In [4-7] the calibration methods with analytical solutions are presented, where in addition to determine internal parameters of the camera, the 3-D object attitude in space are determined as a part of the calibration process. R. Tsai [6] used the Euler angles, while Z. Zhang [5] used Rodrigues' rotation formula.

Researches have been proposed to investigate the measurement of object pose estimation. M. Dhome [8] proposed method to find the analytical solutions to the problem of the determination of the 3-D object attitude in space from a single perspective image. H. Kim [9] proposed a simple and fast stereo matching algorithm for real-time robotic applications using 3D information of vertexes on the outline of an object in image plane. Z. Zhong [10] presents a feature point pair based technique for object pose estimation and 3D structure recovery from a single view, where it is defined strategies for small rotational and large rotational motion, X. Zhang [11] presents algorithms for recovering the camera pose and the 3D-to-2D line correspondences simultaneously.

Determination of angular velocity by image processing is furthermore studied. Y. Zhang [14] by means blurred images processing proposed the estimation of motion parameters by measuring and comparing global geometric properties. S. Wang [13] proposed parameter measurement of rotation through analyzing the information of visual rotation motion blur based on a single blurred image. By using event cameras, which have independent pixels that respond asynchronously to brightness changes, G. Gallego and D. Scaramuzza [12] proposed algorithm to estimate the angular velocity of the camera by analyzing the spatio-temporal coordinates of the brightness change.

Several years ago, it is increased the interest in parameters movement estimation by image processing for space applications. A. Boguslavsky [17] presented a software package that by means of video signal received from the TV-camera, mounted on the spacecraft board, allows the automatic visual monitoring of a spacecraft "Progress" docking to International Space Station. D. Ivanov [18],

proposed a satellite relative position and orientation determination algorithm by performing image processing of the sunlit spacecraft. This algorithm was used to determine the relative movement of the Chibis-M microsatellite developed by IKI RAS.

M. Koptev [19] proposed a method for the translational and rotational motion determination of mock-ups suspended on an aerodynamic testbed. The algorithm was based on the detection of installed special marks on model's body to evaluate the location of model's center of mass, angular position and angular velocity in the coordinate system associated with the aerodynamic testbed.

The difference between the determination algorithm developed and the one described above is that it does not require the installation of an additional special objective or photodiodes on the satellite to shoot; it is enough to know the geometry of the object being shot. The algorithm does not require the transfer of any data from the satellite being taken, therefore a piece of space debris can act as the second device. So, the algorithm is suitable for the tasks of removing space debris from orbit: the satellite companion flies towards the debris, determines its movement, captures it and takes it to the dense atmosphere.

Most of researches on angular motion parameters estimation mentioned above are focused on parameter 3-D object attitude determination or measurement angular velocity, but not both, if 3-D object attitude determination and angular velocity are considered to be estimated at the same time, usually it is considered to add one more sensor in addition to camera.

While in [5] and [6] used Rodrigues' rotation formula and the Euler angles respectively, the purpose of this work is the estimation of the 3-D object attitude by using quaternions, and in addition the angular velocity estimation at the same time by means of a conventional low-cost camera without any additional sensor. The section 1 elaborates the problem statement of this research, and it is explained the importance of measurement model determination, which is the mathematical model for the camera. In the Section 2 it is explained step by step the mathematical expression for the measurement model and its gradient. The linear measurement model derived in the Section 3 allows to determine some parameters of the measurement model by linear methods. Due to the fact that the measurement model h is nonlinear and depends on unknown parameters of the camera, a calibration process is needed to be performed. In the Section 4 the algorithm for calibration is explained and applied for 3-D object attitude determination.

1. Problem statement

The problem of the angular motion determination by the image processing is considered. The source of measurements is the camera which captures the object's movement by taking photographs at a certain frequency (see Figure 1), These pictures are processed to estimate the angular motion of the object using the following information:

- $X_i(x_i, y_i, z_i)$: coordinates of the object points relative to the body-fixed frame OXYZ.
- $X_i'(x_i', y_i')$: coordinates of the same points X_i , which are visualized and located in the image coordinate system (image).

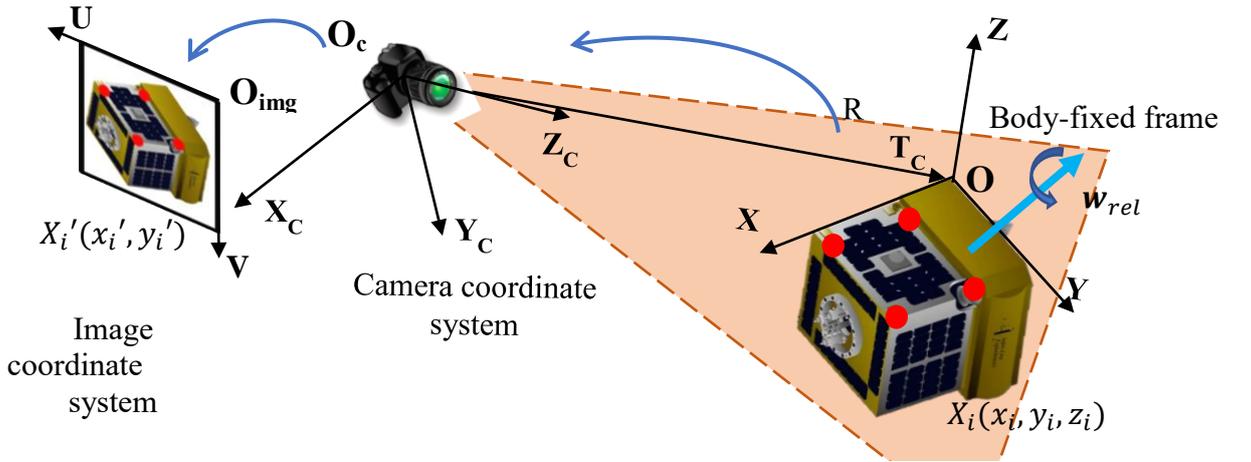


Figure 1. Diagram of the problem statement

The basic structure of a camera is shown in the Figure 2, where two main components are involved:

- Lens: has the function of gathering and focus the light reflected from an object or scene. As the reflected light rays enter the camera lens, they are directed to the image sensor.
- Image sensor: it is a rectangular plane where the points are projected to, representing in that way the image of the object.

The image sensor is located parallel to the lens in the focal plane of the lens. The distance between the lens and the focal plane is called focal length f .

The locations of specific points of an object in an image varies according to the rotation matrix R , and its translation vector T_c with respect to the camera. Thus, the

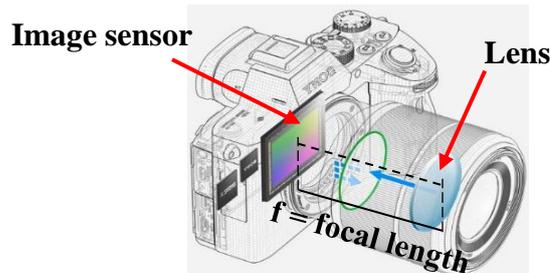


Figure 2. Basic structure of a camera
estimation of the rigid body rotation matrix is possible when function $h(R, T_c)$, called

measurement model, that performs the projection of point X_i into the Image coordinate system is found. In addition, an instantaneous angular velocity can be calculated from two consecutive rotation matrices, thus, the first stage on this work is focused on the rotation matrix determination.

The Cartesian coordinates systems used in this work are shown in the Figure 3:

- $OXYZ$ – Body-fixed Frame (BF) is placed in any location on the object in such a way that the points X_i are known with respect to BF.
- $O_cX_cY_cZ_c$ – Camera Coordinate System (CCS) is based on the pinhole model, where its origin O_c is located at camera center (center of the lens), O_cZ_c is defined by the line from the camera center perpendicular to the image sensor, O_cX_c is parallel to the horizontal side of the image sensor, O_cY_c is parallel to the vertical side of the image sensor.
- $O_{img}UV$ – Image Coordinate System (ICS), also known as the Image plane, is a space of 2D pixel coordinates, where each 2D pixel coordinate is the result of the conversion of points which are located on the image sensor plane in CCS, to 2D coordinate pixel. Its origin O_{img} is located on the top-left corner of the image, $O_{img}U$ extends from left to right and $O_{img}V$ extends downward.

The following notation of points in the different coordinate systems is used:

- $X_i(x_i, y_i, z_i)$ – i -th point with respect to BF.
- $X_{c_i}(x_{c_i}, y_{c_i}, z_{c_i})$ – i -th point with respect to CCS.
- $X'_i(x'_i, y'_i)$ – i -th point with respect to ICS.

With regard to X_i , it is important to notice this point remain fixed with respect to the BF.

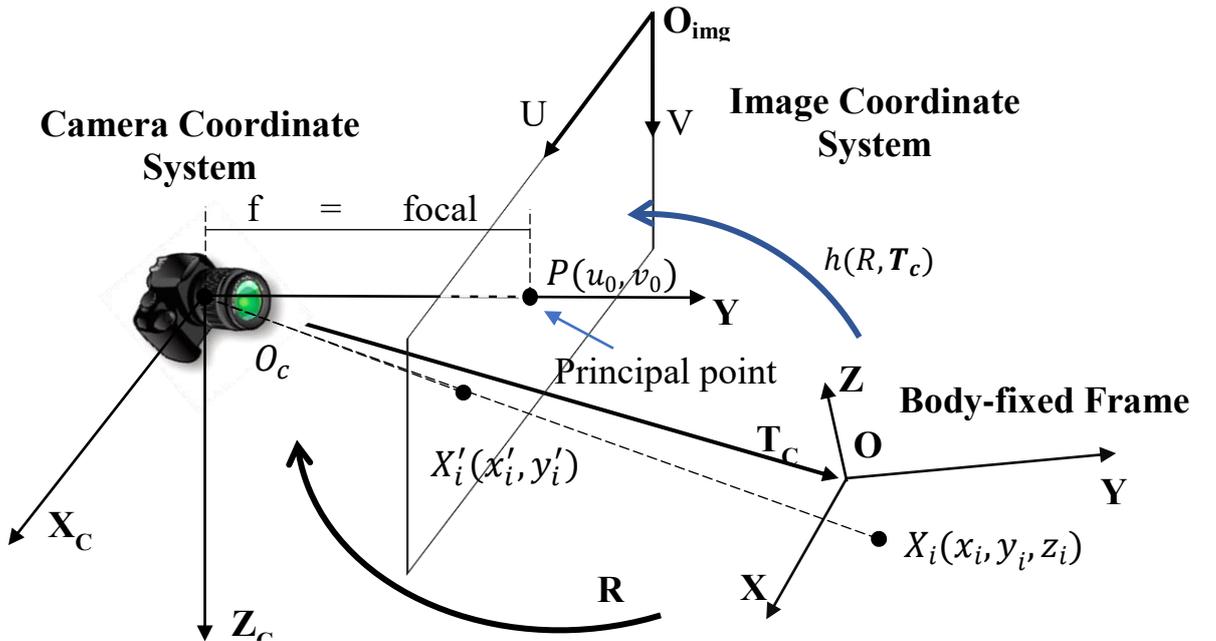


Figure 3. Pinhole camera model

Most of the elements mentioned in this section are considered for measurement model $h(R, T_c)$ definition because of its importance and relevance in the success of

this work, for that reason is given in details the process to define the measurement model.

2. Measurement model

In order to define the measurement model h which is a function that performs the projection of point X_i into the ICS from BF, it is required to consider the following:

- transformation from BF to CCS,
- projection of the points from CCS into the sensor plane,
- lens distortion,
- transformation from sensor plane to ICS

which are going to be explained in details in this section.

2.1. Transformation from BF to CCS

Let $X_i = [x_i, y_i, z_i]^T$ be any point in the BF, where its transformation to the CCS is defined as follows:

$$X_{c_i} = \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \end{bmatrix} = R \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \mathbf{T}_c, \quad (2.1)$$

where the rotation matrix R can be expressed as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.2)$$

and \mathbf{T}_c is translation vector with respect to CCS:

$$\mathbf{T}_c = [t_{xc} \quad t_{yc} \quad t_{zc}]^T. \quad (2.3)$$

From equations (2.1), (2.2) and (2.3) the next expression

$$X_{c_i} = \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \end{bmatrix} = \begin{bmatrix} r_{11}x_i + r_{12}y_i + r_{13}z_i + t_{xc} \\ r_{21}x_i + r_{22}y_i + r_{23}z_i + t_{yc} \\ r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc} \end{bmatrix}$$

performs the transition from points from BF to the CCS.

2.2. Projection of the points from CCS into the image plane

The point X_{cp} represents the projection of the points from CCS into the image plane, and it is expressed as

$$X_{cp_i} = \begin{bmatrix} x_{cp_i} \\ y_{cp_i} \end{bmatrix} = \begin{bmatrix} x_{c_i}/z_{c_i} \\ y_{c_i}/z_{c_i} \end{bmatrix},$$

where

$$x_{cp_i} = \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_{xc}}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc}} \quad (2.4)$$

and

$$y_{cp_i} = \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_{yc}}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_{zc}}. \quad (2.5)$$

It is important to mention that X_{cp} is still located in the CCS.

2.3. Lens distortion

It is necessary to take into account that the lens distortion affects the image during the projection of the point X_c into the sensor plane. The usual types of distortion are radial and tangential ones. Radial distortion can be defined as a function which depends on the distance from the Principal point, this point is formed by the intersection point between the O_cZ_c -axis and the sensor plane. Tangential distortion is caused by an unperfected parallel alignment between the lens and the image sensor [3]. These distortions can be defined by the following expression

$$X_{d_i} = \begin{bmatrix} x_{cp_i}(1 + k_1r_i^2 + k_2r_i^4 + k_3r_i^6) + 2p_1x_{cp_i}y_{cp_i} + p_2(r_i^2 + 2x_{cp_i}^2) \\ y_{cp_i}(1 + k_1r_i^2 + k_2r_i^4 + k_3r_i^6) + 2p_2x_{cp_i}y_{cp_i} + p_1(r_i^2 + 2y_{cp_i}^2) \end{bmatrix},$$

Radial distortion
Tangential distortion

where $X_{d_i} = [x_{d_i}, y_{d_i}]^T$, $r_i^2 = x_{cp_i}^2 + y_{cp_i}^2$ and k_1, k_2, k_3, p_1, p_2 are the distortion coefficients, X_{d_i} is the point coordinates when the lens distortion is taken into account. In case when there is no lens distortion (ideal lens), X_{d_i} and X_{cp_i} are equal.

2.4. Transformation from sensor plane to ICS

Due to the fact that the ICS and sensor plane are parallel, and both located at the same plane, this transformation is based on scaling and translation of the points located in the sensor plane as follows:

$$X_{p_i} = \begin{bmatrix} (x_{d_i} + sy_{d_i})f_x + u_0 \\ y_{d_i}f_y + v_0 \end{bmatrix}, \quad (2.6)$$

where:

- $P = (u_0, v_0)$: Principal point expressed in pixels with respect to ICS.
- $f_x = \alpha_x f$: focal length axis-x (pixel).
- $f_y = \alpha_y f$: focal length axis-y (pixel).
- α_x, α_y : number of pixels per unit distance.

- s : skew coefficient, which usually is equal to zero.
- X_{p_i} : mapped point in the ICS from the BF.
- X_i : point in the BF.

The expression (2.6) can be rewritten as follows:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, R, \mathbf{T}_c), \quad (2.7)$$

where h_{X_i} is called measurement model, which performs the projection of a point X_i into the ICS from the BF. However, taking into account that the parameters $f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2$ are fixed values and specific for each camera, the expression (2.7) can be simplified to $h_{X_i}(R, \mathbf{T}_c)$ once those parameters are determined.

2.5. Measurement model based on Quaternions

Due to the fact that the rotation matrix has 9 scalar elements, it is convenient to express the rotation matrix with less scalar elements during optimization processes. While in [5] and [6] the rotation matrix is expressed using Rodrigues' rotation formula and the Euler angles respectively, in this work the rotation matrix is expressed by means of quaternions.

Taking into account basic quaternion theory, let the multiplication of quaternions $\mathbf{Q} = [q_0, q_1, q_2, q_3]^T$ and $\mathbf{P} = [p_0, p_1, p_2, p_3]^T$ be defined as follow:

$$\mathbf{P} \circ \mathbf{Q} = \begin{bmatrix} p_0 q_0 - \mathbf{p} \mathbf{q} \\ p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \end{bmatrix},$$

where $\mathbf{q} = [q_1, q_2, q_3]^T$ and $\mathbf{p} = [p_1, p_2, p_3]^T$ are vector parts of the quaternions \mathbf{Q} and \mathbf{P} respectively. The previous equation also can be rewritten in a matrix form:

$$\mathbf{P} \circ \mathbf{Q} = \begin{bmatrix} p_0 & -\mathbf{p}^T \\ \mathbf{p} & p_0 I_3 + [\mathbf{p}]_x \end{bmatrix} \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} q_0 & -\mathbf{q}^T \\ \mathbf{q} & q_0 I_3 - [\mathbf{q}]_x \end{bmatrix} \begin{bmatrix} p_0 \\ \mathbf{p} \end{bmatrix} \quad (2.8)$$

The rotation of points by means of quaternions is defined as follows:

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \Lambda \circ \begin{bmatrix} 0 \\ X \end{bmatrix} \circ \tilde{\Lambda}, \quad (2.9)$$

where $X = [x, y, z]^T$ is a tri-dimensional point, and $\Lambda = [\lambda_0, \lambda_1, \lambda_2, \lambda_3]^T$ is an unit quaternion, which modulus $|\Lambda|$ is defined as follows:

$$|\Lambda| = \sqrt{\lambda_0^2 + \lambda_1^2 + \lambda_2^2 + \lambda_3^2} = 1$$

and the conjugated of Λ is represented by

$$\tilde{\Lambda} = \begin{bmatrix} \lambda_0 \\ -\boldsymbol{\lambda} \end{bmatrix}.$$

From the equations (2.9) and (2.8) is obtained the next matrix form for rotation of points:

$$\begin{bmatrix} 0 \\ X_c \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & R(\Lambda) \end{bmatrix} \begin{bmatrix} 0 \\ X \end{bmatrix},$$

where $R(\Lambda)$ represents the rotation matrix as a function of the quaternion Λ . However, due to the fact that Λ is a unit quaternion, the rotation matrix R can be expressed as a function of the vector part $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^T$ of the Λ as follows:

$$R(\boldsymbol{\lambda}) = \begin{bmatrix} 1 - 2(\lambda_2^2 + \lambda_3^2) & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & 1 - 2(\lambda_1^2 + \lambda_3^2) & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & 1 - 2(\lambda_1^2 + \lambda_2^2) \end{bmatrix}. \quad (2.10)$$

Thus, the equation (2.7) can be rewritten as follows:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(f_x, f_y, u_0, v_0, S, k_1, k_2, k_3, p_1, p_2, \boldsymbol{\lambda}, \mathbf{T}_c). \quad (2.11)$$

The measurement model based on quaternion is obtained in the equation (2.11), which is important to determine its gradient for the optimization process.

2.6. Measurement model gradient

Due to the fact that measurement model based on quaternions in the equation (2.11) is a composed function of several transformations, which involves vector and matrices, it is convenient to determine the gradient by means of matrix calculus. Let the measurement model be rewritten as follow:

$$X_{p_i} = \begin{bmatrix} x_{p_i} \\ y_{p_i} \end{bmatrix} = h_{X_i}(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \boldsymbol{\lambda}, \mathbf{T}_c), \quad (2.12)$$

where $\mathbf{F} = [f_x \ f_y]$, $\mathbf{C} = [u_0 \ v_0]$, and $\mathbf{K}_D = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]$, and let the measurement model gradient be defined as follows:

$$Dh_{X_i} = \begin{bmatrix} \frac{\partial h_{X_i}}{\partial \mathbf{F}} & \frac{\partial h_{X_i}}{\partial \mathbf{C}} & \frac{\partial h_{X_i}}{\partial S} & \frac{\partial h_{X_i}}{\partial \mathbf{K}_D} & \frac{\partial h_{X_i}}{\partial \boldsymbol{\lambda}} & \frac{\partial h_{X_i}}{\partial \mathbf{T}_c} \end{bmatrix}, \quad (2.13)$$

where $\partial h_{X_i}/\partial \mathbf{F}$, $\partial h_{X_i}/\partial \mathbf{C}$, $\partial h_{X_i}/\partial S$, and $\partial h_{X_i}/\partial \mathbf{K}_D$ are defined in a matrix form:

$$\frac{\partial h_{X_i}}{\partial \mathbf{F}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \mathbf{F}} \\ \frac{\partial y_{p_i}}{\partial \mathbf{F}} \end{bmatrix} = \begin{bmatrix} x_{d_i} + sy_{d_i} & 0 \\ 0 & y_{d_i} \end{bmatrix} \quad (2.14)$$

$$\frac{\partial h_{X_i}}{\partial \mathbf{c}} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \mathbf{c}} \\ \frac{\partial y_{p_i}}{\partial \mathbf{c}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{\partial h_{X_i}}{\partial s} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial s} \\ \frac{\partial y_{p_i}}{\partial s} \end{bmatrix} = \begin{bmatrix} y_{p_i} f_x \\ 0 \end{bmatrix}$$

$$\frac{\partial h_{X_i}}{\partial \mathbf{K}_D} = \begin{bmatrix} x_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2) & x_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2)^2 & 2x_{d_i}y_{d_i} & y_{cp_i}^2 + 3x_{cp_i}^2 & x_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2)^3 \\ y_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2) & y_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2)^2 & x_{cp_i}^2 + 3y_{cp_i}^2 & 2x_{d_i}y_{d_i} & y_{cp_i}(x_{cp_i}^2 + y_{cp_i}^2)^3 \end{bmatrix}$$

With regard to $\partial h_{X_i}/\partial \boldsymbol{\lambda}$ and $\partial h_{X_i}/\partial \mathbf{T}_c$, these values have remarkably complicated expressions due to the fact that the measurement model in the equation (2.11) is a composed function of several transformations.

As the rotation matrix R is a function of $\boldsymbol{\lambda}$, see equation (2.10), the derivative $\partial h_{X_i}/\partial \boldsymbol{\lambda}$ is

$$\frac{\partial h_{X_i}}{\partial \boldsymbol{\lambda}} = \frac{\partial h_{X_i}}{\partial R} \frac{\partial R(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}, \quad (2.15)$$

$$\frac{\partial h_{X_i}}{\partial R} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial R} \\ \frac{\partial y_{p_i}}{\partial R} \end{bmatrix},$$

where $\partial y_{p_i}/\partial R$ and $\partial x_{p_i}/\partial R$ are

$$\frac{\partial y_{p_i}}{\partial R} = f_y \left(D_{r_i} \frac{\partial y_{cp_i}}{\partial R} + y_{cp_i} \left[k_1 \frac{\partial(r_i^2)}{\partial R} + k_2 \frac{\partial(r_i^4)}{\partial R} + k_3 \frac{\partial(r_i^6)}{\partial R} \right] + (2p_2 y_{cp_i} + 2p_1 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial R} + (2p_2 x_{cp_i} + 6p_1 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial R} \right),$$

$$\frac{\partial x_{p_i}}{\partial R} = f_x \left(D_{r_i} \frac{\partial x_{cp_i}}{\partial R} + x_{cp_i} \left[k_1 \frac{\partial(r_i^2)}{\partial R} + k_2 \frac{\partial(r_i^4)}{\partial R} + k_3 \frac{\partial(r_i^6)}{\partial R} \right] + (2p_1 y_{cp_i} + 6p_2 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial R} + (2p_1 x_{cp_i} + 2p_2 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial R} + s f_x \left(D_{r_i} \frac{\partial y_{cp_i}}{\partial R} + y_{cp_i} \left[k_1 \frac{\partial(r_i^2)}{\partial R} + k_2 \frac{\partial(r_i^4)}{\partial R} + k_3 \frac{\partial(r_i^6)}{\partial R} \right] + (2p_2 y_{cp_i} + 2p_1 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial R} + (2p_2 x_{cp_i} + 6p_1 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial R} \right), \right.$$

$$\frac{\partial(r_i^2)}{\partial R} = 2[x_{cp_i} \quad y_{cp_i}] \frac{\partial \mathbf{X}_{cp_i}}{\partial R},$$

$$\frac{\partial(r_i^6)}{\partial R} = 3r_i^4 \frac{\partial(r_i^2)}{\partial R} = 6 \left(x_{cp_i}^2 + y_{cp_i}^2 \right)^2 [x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial R}.$$

The expression for $\partial R/\partial \lambda$ is

$$\frac{\partial R}{\partial \lambda} = \begin{bmatrix} 0 & -4\lambda_2 & -4\lambda_3 \\ 2\lambda_2 - 2\lambda_3\lambda_1/\lambda_0 & 2\lambda_1 - 2\lambda_3\lambda_2/\lambda_0 & 2\lambda_0 - 2\lambda_3\lambda_3/\lambda_0 \\ 2\lambda_3 + 2\lambda_2\lambda_1/\lambda_0 & -2\lambda_0 + \lambda_2\lambda_2/\lambda_0 & 2\lambda_1 + 2\lambda_2\lambda_3/\lambda_0 \\ 2\lambda_2 + 2\lambda_3\lambda_1/\lambda_0 & 2\lambda_1 + 2\lambda_3\lambda_2/\lambda_0 & -2\lambda_0 + 2\lambda_3\lambda_3/\lambda_0 \\ -4\lambda_1 & 0 & -4\lambda_3 \\ 2\lambda_0 - 2\lambda_1\lambda_1/\lambda_0 & 2\lambda_3 - 2\lambda_1\lambda_2/\lambda_0 & 2\lambda_2 - 2\lambda_1\lambda_3/\lambda_0 \\ 2\lambda_3 - 2\lambda_2\lambda_1/\lambda_0 & 2\lambda_0 - 2\lambda_2\lambda_2/\lambda_0 & 2\lambda_1 - 2\lambda_2\lambda_3/\lambda_0 \\ -2\lambda_0 + 2\lambda_1\lambda_1/\lambda_0 & 2\lambda_3 + 2\lambda_1\lambda_2/\lambda_0 & 2\lambda_2 + 2\lambda_1\lambda_3/\lambda_0 \\ -4\lambda_1 & -4\lambda_2 & 0 \end{bmatrix}.$$

The derivative $\partial h_{X_i}/\partial \mathbf{T}_c$ is

$$\frac{\partial h_{X_i}}{\partial \mathbf{T}_c} = \begin{bmatrix} \frac{\partial x_{p_i}}{\partial \mathbf{T}_c} \\ \frac{\partial y_{p_i}}{\partial \mathbf{T}_c} \end{bmatrix}, \quad (2.16)$$

$$\begin{aligned} \frac{\partial x_{p_i}}{\partial \mathbf{T}_c} = & f_x \left(D_{r_i} \frac{\partial x_{cp_i}}{\partial \mathbf{T}_c} + x_{cp_i} \left[k1 \frac{\partial(r_i^2)}{\partial \mathbf{T}_c} + k2 \frac{\partial(r_i^4)}{\partial \mathbf{T}_c} + k3 \frac{\partial(r_i^6)}{\partial \mathbf{T}_c} \right] + (2p_1 y_{cp_i} + 6p_2 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial \mathbf{T}_c} + \right. \\ & \left. (2p_1 x_{cp_i} + 2p_2 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} \right) + sf_x \left(D_{r_i} \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} + y_{cp_i} \left[k1 \frac{\partial(r_i^2)}{\partial \mathbf{T}_c} + k2 \frac{\partial(r_i^4)}{\partial \mathbf{T}_c} + k3 \frac{\partial(r_i^6)}{\partial \mathbf{T}_c} \right] + \right. \\ & \left. (2p_2 y_{cp_i} + 2p_1 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial \mathbf{T}_c} + (2p_2 x_{cp_i} + 6p_1 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} \right), \end{aligned}$$

$$\begin{aligned} \frac{\partial y_{p_i}}{\partial \mathbf{T}_c} = & f_y \left(D_{r_i} \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} + y_{cp_i} \left[k1 \frac{\partial(r_i^2)}{\partial \mathbf{T}_c} + k2 \frac{\partial(r_i^4)}{\partial \mathbf{T}_c} + k3 \frac{\partial(r_i^6)}{\partial \mathbf{T}_c} \right] + (2p_2 y_{cp_i} + 2p_1 x_{cp_i}) \frac{\partial x_{cp_i}}{\partial \mathbf{T}_c} + \right. \\ & \left. (2p_2 x_{cp_i} + 6p_1 y_{cp_i}) \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} \right), \end{aligned}$$

$$\frac{\partial(r_i^2)}{\partial \mathbf{T}_c} = 2[x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \mathbf{T}_c},$$

$$\frac{\partial(r_i^4)}{\partial \mathbf{T}_c} = 2r_i^2 \frac{\partial(r_i^2)}{\partial \mathbf{T}_c} = 4 \left(x_{cp_i}^2 + y_{cp_i}^2 \right) [x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \mathbf{T}_c},$$

$$\frac{\partial(r_i^6)}{\partial \mathbf{T}_c} = 3r_i^4 \frac{\partial(r_i^2)}{\partial \mathbf{T}_c} = 6 \left(x_{cp_i}^2 + y_{cp_i}^2 \right)^2 [x_{cp_i} \quad y_{cp_i}] \frac{\partial X_{cp_i}}{\partial \mathbf{T}_c},$$

$$\frac{\partial X_{cp_i}}{\partial \mathbf{T}_c} = \begin{bmatrix} \frac{\partial x_{cp_i}}{\partial \mathbf{T}_c} \\ \frac{\partial y_{cp_i}}{\partial \mathbf{T}_c} \end{bmatrix} = \begin{bmatrix} \frac{1}{z_{c_i}} & 0 & -\frac{x_{c_i}}{z_{c_i}^2} \\ 0 & \frac{1}{z_{c_i}} & -\frac{y_{c_i}}{z_{c_i}^2} \end{bmatrix}.$$

It is important to mention that the measurement model gradient is obtained for any point X_i located in BF and projected to ICS. And as a result, it is obtained a 2×16 matrix for each point X_i , which is taken into account in the optimization process.

3. Linearization of the measurement model

The linearization of the measurement model allows to solve the nonlinear calibration problem for cameras by linear method, in which the nonlinear radial and tangential distortion components are ignored.

The expressions (2.4) and (2.5) are nonlinear functions, which perform the projection to the sensor plane, and can be linearized by means of Homogeneous coordinates provided that the vector X_{c_i} and X_{cp_i} are expressed homogeneous vectors [1], obtaining the equation (3.1) as a linear expression, where the symbol \sim means that the two homogeneous vectors are not equal, but they have the same direction.

$$\begin{bmatrix} x_{cp_i} \\ y_{cp_i} \\ z_{cp_i} \end{bmatrix} \sim \begin{bmatrix} x_{c_i} f \\ y_{c_i} f \\ z_{c_i} \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{c_i} \\ y_{c_i} \\ z_{c_i} \\ 1 \end{bmatrix} \quad (3.1)$$

With regard to the effect of the lens distortion, it is convenient to consider it to be equal to zero during the linearization process [2]. Therefore, considering this particular case it is possible to obtain a linear expression, see equation (3.2), from the nonlinear measurement model (2.6) by means of the homogeneous coordinates which is usually done in order to determine initial values of internal and external parameter.

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \sim \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \mathbf{T}_c \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (3.2)$$

In the equation (3.2), \tilde{u} , \tilde{v} , \tilde{w} are homogeneous coordinates for the points in the ICS, and can be expressed as follows:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \sim [H_{3 \times 4}] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \quad (3.3)$$

where matrix H is the transition matrix, or linear measurement model. The BF is chosen in such a way that the points X_i are located on the XY-plane, in consequence, the component z_i is zero, it means that the equation (3.3) can be reduced to equation (3.4).

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \tilde{w}_i \end{bmatrix} \sim \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3.4)$$

Due to the fact that the vectors $[\tilde{u}_i, \tilde{v}_i, \tilde{w}_i]^T$ and $H[x_i, y_i, 1]^T$ have the same direction, their cross product is zero and based on the Direct Linear Transformation (DLT) algorithm [1] the equation is

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \tilde{w}_i \end{bmatrix} \times H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 0^T & -\tilde{w}_i X_i^T & \tilde{v}_i X_i^T \\ \tilde{w}_i X_i^T & 0^T & -\tilde{u}_i X_i^T \end{bmatrix} \mathbf{L} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (3.5)$$

where $\mathbf{L} = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_8 \ a_9]^T$ and $X_i = [x_i \ y_i \ 1]^T$.

As it can be seen, the equation (3.5) has the form of a homogeneous system, where \mathbf{L} can be determined by the Single Values Decomposition (SVD). This DLT algorithm is widely used to calculate the transition matrix H where is needed a set of four points as minimum. However, because matrix H is a projective transformation, it has a non-linear nature, therefore, an iterative method can be applied in order to optimize the components of the matrix H by means of reduction of the error projection [2]. Thus, it is necessary to work in inhomogeneous coordinates.

Let the matrix H already be determined by means of DLT, then

$$\begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ \tilde{\omega}_i \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad (3.6)$$

where $(\tilde{x}_i, \tilde{y}_i, \tilde{\omega}_i)$ is the homogeneous coordinate representation of a point (u_i, v_i) located in the ICS, then the projective transformation in the equation (3.6) can be written in inhomogeneous form as

$$u_i = \frac{\tilde{x}_i}{\tilde{\omega}_i} = \frac{a_1 x_i + a_2 y_i + a_3}{a_7 x_i + a_8 y_i + a_9}, \quad (3.7)$$

$$v_i = \frac{\tilde{y}_i}{\tilde{\omega}_i} = \frac{a_4 x_i + a_5 y_i + a_6}{a_7 x_i + a_8 y_i + a_9}, \quad (3.8)$$

where (u_i, v_i) finally represents the mapped point in the ICS from the BF. The Jacobian matrix for projective transformation is shown below, which is widely used by the most of the iterative methods.

$$J = \frac{\partial \begin{bmatrix} u_i \\ v_i \end{bmatrix}}{\partial \mathbf{L}} = \frac{1}{\tilde{w}_i} \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i & -u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_i x_i & -v_i y_i & -v_i \end{bmatrix} \quad (3.9)$$

Below it is described an algorithm to compute the transition matrix H .

I. Initialize data:

- Let $i = 1, 2, \dots, n$, where $n \geq 4$ is the number of mapped points.
- Let $X_i = [x_i \ y_i \ 1]^T$ be a homogeneous coordinate representation of a i -th point from the BF, where the component z_i is zero.
- Let $(\tilde{u}_i, \tilde{v}_i, \tilde{w}_i)$ be a homogeneous coordinate representation of a i -th point located in the ICS.
- Let \tilde{w}_i to be one, in order to make $(\tilde{u}_i, \tilde{v}_i)$ points measured in the ICS.

✓ Apply the preconditioning matrix to each point as follow:

$$\begin{bmatrix} \tilde{u}_i' \\ \tilde{v}_i' \\ \tilde{w}_i' \end{bmatrix} = H_{prec} \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \tilde{w}_i \end{bmatrix}$$

- Write the homogeneous system according to the equation (3.5) for n points.
- Solve the homogenous system in order to obtain L , and obtain the transition matrix H from L .
- Update the transition matrix as follow:

$$\begin{aligned} H &\leftarrow H/a_9 \\ H &\leftarrow H_{prec}^{-1}H \end{aligned}$$

II. Optimize the matrix H :

- Let $X_i' = [x_i', y_i']^T$ be measured point in the ICS.
- Let $P_i = [u_i, v_i]^T$ mapped point in the image coordinate obtained from the equations (3.7) and (3.8).
- By means of iterative process and using the Jacobian matrix of the equation (3.8), minimize:

$$\epsilon = \sum \|X_i' - P_i\|$$

Algorithm 1. Computing Transition matrix H

Before determining the transition matrix, a normalization of the data is recommended to avoid bad results because of noisy data. In [1] is recommended a normalization data so that the centroid of the new set of points is the origin of coordinates (0,0) and the average distance from the origin equals to $\sqrt{2}$.

In the next section it is shown how the Computing transition matrix algorithm can be used to estimate the rotation matrix and translation vector of the BF with respect to CCS by a linear method rather than use the linearized measurement model.

4. Calibration Algorithm

In this section the calibration algorithm of the widely known tool for camera calibration developed in [2], based on Rodrigues' rotation formula, is adapted to quaternion rotation representation.

In the Figure 4, it is shown that a chessboard is photographed with different orientations and translation vectors in order to obtain considerable amount of points for calibration process. Additionally, intrinsic parameters are shown, which are internal fixed parameters of the camera itself. They have to be determined in the calibration process and then will remain fixed. On the other hand, extrinsic parameters, rotation matrix and translation vector, are determined for each image, and they are not fixed parameters because the location and orientation of the object can

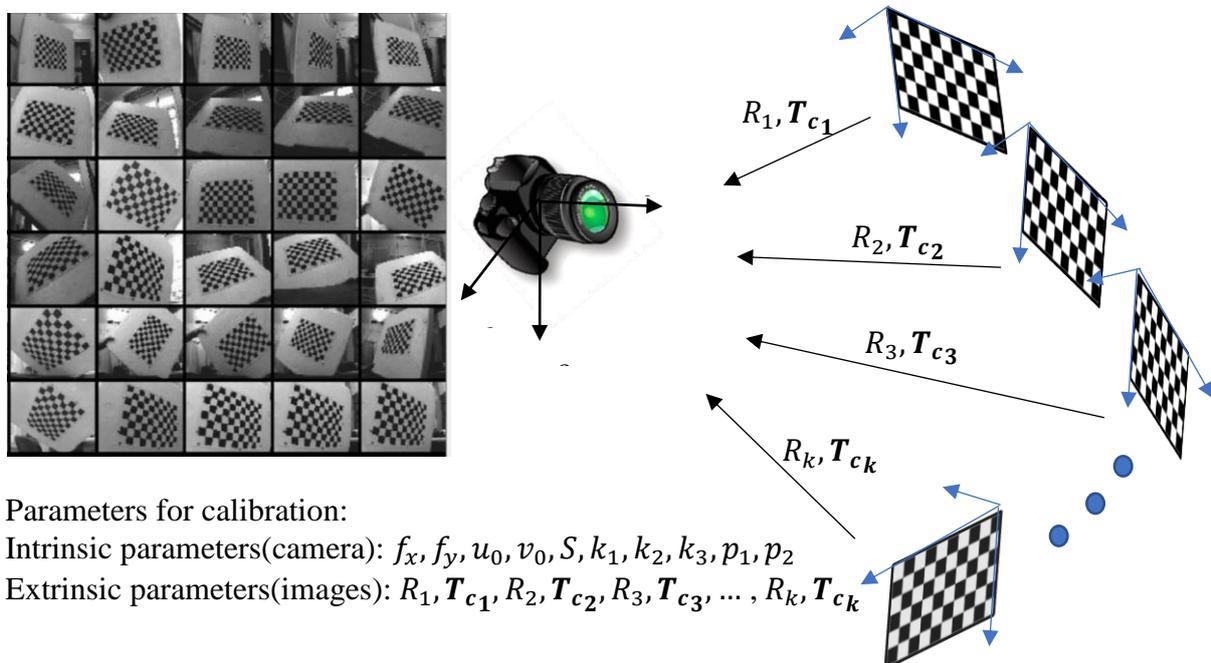


Figure 4. Intrinsic and Extrinsic parameters

change.

Calibration process is based on two main steps: initialization of the parameters and optimization of the parameters by the gradient method.

4.1. Initialization of parameters

The initial value of the principal point can be initialized as the center point of the image, for example, if the resolution of the camera is 640x480 pixels, then the principal point $P = (u_0, v_0) = (320, 240)$. The Skew parameter can be initialized as zero as well as the distortions coefficients k_1, k_2, k_3, p_1, p_2 .

With regards to the focal distance (f_x, f_y) , it can be initialized using vanishing points as in [1] and different methods as in [2] and [4], which make use of transition matrices from BF to the ICS by using the Algorithm 1.

Considering initial values for skew factor ‘S’ and distortion coefficients K_D equal to zero, the points in the ICS (x'_i, y'_i) can be transformed into CCS as follows:

$$x_{cp_i} = (x'_i - u_0)/f_x,$$

$$y_{cp_i} = (y'_i - v_0)/f_y.$$

The equations above show that the point (x'_i, y'_i) is located in the sensor plane in the CCS, and it is related to the BF by the next equation, where the points in the ICS and BF are expressed by homogeneous coordinate.

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} R & T_c \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \quad (4.1)$$

Because of component z_i is zero for flat objects, equation (4.1) can be rewritten as

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \sim [\mathbf{r}_1 \quad \mathbf{r}_2 \quad T_c] \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = H_r \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix},$$

where T_c is translation vector and \mathbf{r}_i are the columns of rotation matrix R, and $\|\mathbf{r}_i\| = 1$. The matrix H_r can be computed by means of the Algorithm 1, and additionally it is necessary to perform a normalization so that the vectors \mathbf{r}_i have modulus equal to one, then to use the QR decomposition to obtain a better result in the orthogonality of the vector \mathbf{r}_i .

4.2. Optimization process

Due to the non-linearity of the measurement model, an optimization process is required to be performed in order to tune up the parameters which have been initialized previously. The essential step is the definition of the equations.

Let us consider a scheme where it is available just one image as it is shown in the Figure 5. Let $i = 1, 2, \dots, n$, where n is the number of mapped points to the ICS. Let $X'_i = [x'_i, y'_i]^T$ and $X_i = [x_i, y_i, 0]^T$ be the known vector representations of a point in the ICS and BF.

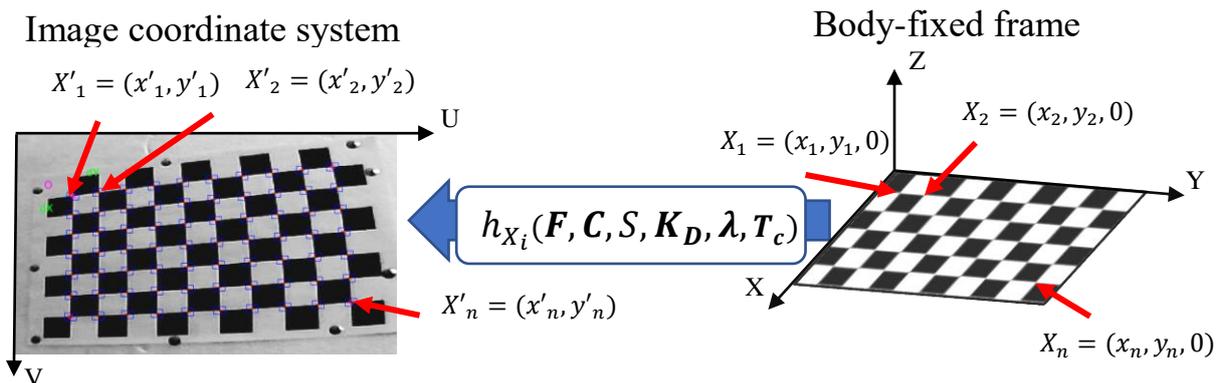


Figure 5. Projection from Body-fixed frame to image coordinate system

Let $Xp_i = [xp_i, yp_i]^T$ be i -th point already mapped to the ICS from the BF by using the measurement model h_{x_i} from the equation (2.12), where the rotation matrix can be expressed by using the vector part λ of a unit quaternion, below the equations for one image with n points:

$$\begin{aligned} Xp_1 &= h_{x_1}(F, C, S, K_D, \lambda, T_c) \\ Xp_2 &= h_{x_2}(F, C, S, K_D, \lambda, T_c) \\ &\vdots \\ Xp_i &= h_{x_i}(F, C, S, K_D, \lambda, T_c) \\ &\vdots \\ Xp_n &= h_{x_n}(F, C, S, K_D, \lambda, T_c) \end{aligned}$$

Let ΔX be the error vector which is defined as the difference between the points Xp_i and X'_i as follow:

$$\Delta X = \begin{bmatrix} Xp_1 - X'_1 \\ Xp_2 - X'_2 \\ \vdots \\ Xp_n - X'_n \end{bmatrix} = \begin{bmatrix} xp_1 - x'_1 \\ yp_1 - y'_1 \\ xp_2 - x'_2 \\ yp_2 - y'_2 \\ \vdots \\ xp_n - x'_n \\ yp_n - y'_n \end{bmatrix}. \quad (4.2)$$

Now let us assume that m images are available with n points in each image. Let $i = 1, 2, \dots, n$, and $k = 1, 2, \dots, m$ where n is the number of mapped points to the ICS and m is the number images. It is important to mention that the location of points in each image depends on the translation vector and orientation of the BF with respect to the CCS.

Let $X'_i^k = [x'_i^k, y'_i^k]^T$ be the vector representation of the i -th point in the k -th image (ICS).

Let $X_i = [x_i, y_i, 0]^T$ be the vector representation of the i -th point in the BF.

Let $Xp_i^k = [xp_i^k, yp_i^k]^T$ be the point X_i already mapped to the k -th image (ICS) from the BF by using the nonlinear model $h_{x_i}^k$, which represent the projection of the point X_i to the k -th image. Below the equations for m images with n points in each image are given:

$$\begin{aligned}
Xp_1^1 &= h_{X_1}^1(F, C, S, K_D, \lambda_1, T_{c_1}) \\
Xp_2^1 &= h_{X_2}^1(F, C, S, K_D, \lambda_1, T_{c_1}) \\
&\vdots \\
Xp_n^1 &= h_{X_n}^1(F, C, S, K_D, \lambda_1, T_{c_1}) \\
Xp_1^2 &= h_{X_1}^2(F, C, S, K_D, \lambda_2, T_{c_2}) \\
Xp_2^2 &= h_{X_2}^2(F, C, S, K_D, \lambda_2, T_{c_2}) \\
&\vdots \\
Xp_n^2 &= h_{X_n}^2(F, C, S, K_D, \lambda_2, T_{c_2}) \\
&\vdots \\
Xp_i^k &= h_{X_i}^k(F, C, S, K_D, \lambda_k, T_{c_k}) \\
&\vdots \\
Xp_1^m &= h_{X_1}^m(F, C, S, K_D, \lambda_m, T_{c_m}) \\
Xp_2^m &= h_{X_2}^m(F, C, S, K_D, \lambda_m, T_{c_m}) \\
&\vdots \\
Xp_n^m &= h_{X_n}^m(F, C, S, K_D, \lambda_m, T_{c_m})
\end{aligned} \tag{4.3}$$

Let h^k be defined as

$$h^k(F, C, S, K_D, \lambda_k, T_{c_k}) = \begin{bmatrix} h_{X_1}^k(F, C, S, K_D, \lambda_k, T_{c_k}) \\ h_{X_2}^k(F, C, S, K_D, \lambda_k, T_{c_k}) \\ \vdots \\ h_{X_n}^k(F, C, S, K_D, \lambda_k, T_{c_k}) \end{bmatrix}.$$

The partial derivatives of h^k from the measurement model gradient in the equation (2.13) are

$$\left[\frac{\partial h^k}{\partial F} \quad \frac{\partial h^k}{\partial C} \quad \frac{\partial h^k}{\partial S} \quad \frac{\partial h^k}{\partial K_D} \quad \frac{\partial h^k}{\partial \lambda_k} \quad \frac{\partial h^k}{\partial T_{c_k}} \right] = \begin{bmatrix} \frac{\partial h_{X_1}^k}{\partial F} & \frac{\partial h_{X_1}^k}{\partial C} & \frac{\partial h_{X_1}^k}{\partial S} & \frac{\partial h_{X_1}^k}{\partial K_D} & \frac{\partial h_{X_1}^k}{\partial \lambda_k} & \frac{\partial h_{X_1}^k}{\partial T_{c_k}} \\ \frac{\partial h_{X_2}^k}{\partial F} & \frac{\partial h_{X_2}^k}{\partial C} & \frac{\partial h_{X_2}^k}{\partial S} & \frac{\partial h_{X_2}^k}{\partial K_D} & \frac{\partial h_{X_2}^k}{\partial \lambda_k} & \frac{\partial h_{X_2}^k}{\partial T_{c_k}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_n^k}{\partial F} & \frac{\partial h_n^k}{\partial C} & \frac{\partial h_n^k}{\partial S} & \frac{\partial h_n^k}{\partial K_D} & \frac{\partial h_n^k}{\partial \lambda_k} & \frac{\partial h_n^k}{\partial T_{c_k}} \end{bmatrix},$$

then the equations (4.3) can be expressed in a shorter form as follows:

$$\begin{aligned}
Xp^1 &= h^1(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_1, \mathbf{T}_{c_1}) \rightarrow \Delta X^1 = Xp^1 - X'^1, \\
Xp^2 &= h^2(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_2, \mathbf{T}_{c_2}) \rightarrow \Delta X^2 = Xp^2 - X'^2, \\
&\vdots \\
Xp^k &= h^k(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_k, \mathbf{T}_{c_k}) \rightarrow \Delta X^k = Xp^k - X'^k, \\
&\vdots \\
Xp^m &= h^m(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_m, \mathbf{T}_{c_m}) \rightarrow \Delta X^m = Xp^m - X'^m
\end{aligned}$$

where $Xp^k = [xp_1^k, yp_1^k, xp_2^k, yp_2^k, \dots, xp_n^k, yp_n^k]^T$ is column vector representation of the points mapped to the k -th image (ICS) from the BF, and $X'^k = [x'_1, y'_1, x'_2, y'_2, \dots, x'_n, y'_n]^T$ is vector representation of all the points in the k -th image.

In the equation (4.2) the error vector ΔX can be express as ΔX^k where k indicates the error vector for the corresponding k -th image and nonlinear model h^k .

Finally, the equations can be express as a column of functions

$$\hat{h}(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_1, \mathbf{T}_{c_1}, \dots, \lambda_m, \mathbf{T}_{c_m}) = \begin{bmatrix} Xp^1 \\ Xp^2 \\ \vdots \\ Xp^m \end{bmatrix} = \begin{bmatrix} h^1(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_1, \mathbf{T}_{c_1}) \\ h^2(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_2, \mathbf{T}_{c_2}) \\ \vdots \\ h^m(\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_m, \mathbf{T}_{c_m}) \end{bmatrix}$$

Let X' be the column vector $[X'^1, X'^2, \dots, X'^m]^T$, and the global error vector can be defined as follows:

$$\epsilon = \hat{h}(M) - X' = \begin{bmatrix} \Delta X^1 \\ \Delta X^2 \\ \vdots \\ \Delta X^m \end{bmatrix}.$$

The Gauss–Newton Method is used to solve the optimization problem, which is based on the minimization of the global error vector ϵ ; Let $\mathbf{M} = [\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_1, \mathbf{T}_{c_1}, \dots, \lambda_m, \mathbf{T}_{c_m}]$ be the vector of parameters, let \mathbf{M}_0 be the initial values for the vector of parameters \mathbf{M} , and let ϵ_0 be the initial error vector.

$$\epsilon_0 = \hat{h}(\mathbf{M}_0) - X'$$

Let ϵ_l be the error vector and \mathbf{M}_l be the vector of parameters which are updated for each iteration as follows:

$$\epsilon_l = \hat{h}(\mathbf{M}_l) - X',$$

$$\Delta \mathbf{M} = (J^T J)^{-1} J^T \epsilon_l,$$

$$\mathbf{M}_{l+1} = \mathbf{M}_l + \Delta \mathbf{M}$$

where the Jacobian matrix J is defined as follows:

$$J = \frac{\partial \mathbf{h}}{\partial \mathbf{M}} = \begin{bmatrix} \frac{\partial h^1}{\partial \mathbf{M}} \\ \frac{\partial h^2}{\partial \mathbf{M}} \\ \frac{\partial h^3}{\partial \mathbf{M}} \\ \vdots \\ \frac{\partial h^k}{\partial \mathbf{M}} \\ \vdots \\ \frac{\partial h^m}{\partial \mathbf{M}} \end{bmatrix} = \begin{bmatrix} \frac{\partial h^1}{\partial \mathbf{F}} & \frac{\partial h^1}{\partial \mathbf{C}} & \frac{\partial h^1}{\partial S} & \frac{\partial h^1}{\partial \mathbf{K}_D} & \frac{\partial h^1}{\partial \lambda_1} & \frac{\partial h^1}{\partial \mathbf{T}_{c_1}} & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{\partial h^2}{\partial \mathbf{F}} & \frac{\partial h^2}{\partial \mathbf{C}} & \frac{\partial h^2}{\partial S} & \frac{\partial h^2}{\partial \mathbf{K}_D} & 0 & 0 & \frac{\partial h^2}{\partial \lambda_2} & \frac{\partial h^2}{\partial \mathbf{T}_{c_2}} & 0 & 0 & \dots & 0 \\ \frac{\partial h^3}{\partial \mathbf{F}} & \frac{\partial h^3}{\partial \mathbf{C}} & \frac{\partial h^3}{\partial S} & \frac{\partial h^3}{\partial \mathbf{K}_D} & 0 & 0 & 0 & 0 & \frac{\partial h^3}{\partial \lambda_3} & \frac{\partial h^3}{\partial \mathbf{T}_{c_3}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \frac{\partial h^k}{\partial \mathbf{F}} & \frac{\partial h^k}{\partial \mathbf{C}} & \frac{\partial h^k}{\partial S} & \frac{\partial h^k}{\partial \mathbf{K}_D} & 0 & 0 & 0 & \dots & 0 & \frac{\partial h^k}{\partial \lambda_k} & \frac{\partial h^k}{\partial \mathbf{T}_{c_k}} & 0 \\ \vdots & \ddots & \vdots \\ \frac{\partial h^m}{\partial \mathbf{F}} & \frac{\partial h^m}{\partial \mathbf{C}} & \frac{\partial h^m}{\partial S} & \frac{\partial h^m}{\partial \mathbf{K}_D} & 0 & 0 & 0 & 0 & \dots & 0 & \frac{\partial h^m}{\partial \lambda_m} & \frac{\partial h^m}{\partial \mathbf{T}_{c_m}} \end{bmatrix}$$

where $\mathbf{F} = [f_x \ f_y]$, $\mathbf{C} = [u_0 \ v_0]$ and $\mathbf{K}_D = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]$.

4.3. Algorithm for camera calibration

Assume m images with n points in each image are given and let $i = 1, 2, \dots, n$, and $k = 1, 2, \dots, m$ where n is the number of mapped points to the ICS and m is the number images. The calibration algorithm is shown below.

- I. **Initialize parameters:** Use the Algorithm 1 to initialize the vector parameters $\mathbf{M} = [\mathbf{F}, \mathbf{C}, S, \mathbf{K}_D, \lambda_1, \mathbf{T}_{c_1}, \dots, \lambda_m, \mathbf{T}_{c_m}]$.
- II. **Initialize global error vector ϵ_0 :** $\epsilon_0 = \mathbf{h}(\mathbf{M}_0) - X'$
- III. **Iterative process:**
 - a. $\Delta \mathbf{M}_0 = (J_0^T J_0)^{-1} J_0^T \epsilon_0$, where J_0 is J jacobian matrix evaluated at \mathbf{M}_0
 - b. $\mathbf{M}_1 = \mathbf{M}_0 + \Delta \mathbf{M}_0$
 - c. Change $\leftarrow |\Delta \mathbf{M}_0| / |\mathbf{M}_1|$
 - d. Iteration $\leftarrow 0$
 - e. While ((Change > 1e-10) & (Iteration < MaxIteration))
 - i. $\epsilon_1 = \mathbf{h}(\mathbf{M}_1) - X'$
 - ii. $\Delta \mathbf{M}_1 = (J_1^T J_1)^{-1} J_1^T \epsilon_1$
 - iii. $\mathbf{M}_2 = \mathbf{M}_1 + \Delta \mathbf{M}_1$
 - iv. The quaternion part of \mathbf{M}_2 must be normalized for each iteration, and then \mathbf{M}_2 must be updated.
 - v. Change $\leftarrow |\Delta \mathbf{M}_1| / |\mathbf{M}_2|$
 - vi. Iteration \leftarrow Iteration + 1

The vector \mathbf{M}_2 is the optimal vector \mathbf{M} .

Algorithm 2. Camera calibration

After the calibration the Algorithm 2 can be used to determine the matrix rotation and \mathbf{T}_c without considering the other parameters in the vector \mathbf{M} .

4.4. Application algorithm for camera calibration

Using 50 images and 70 points per image. The images were taken using the camera Model FI8918W with resolution 480x640 pixels.

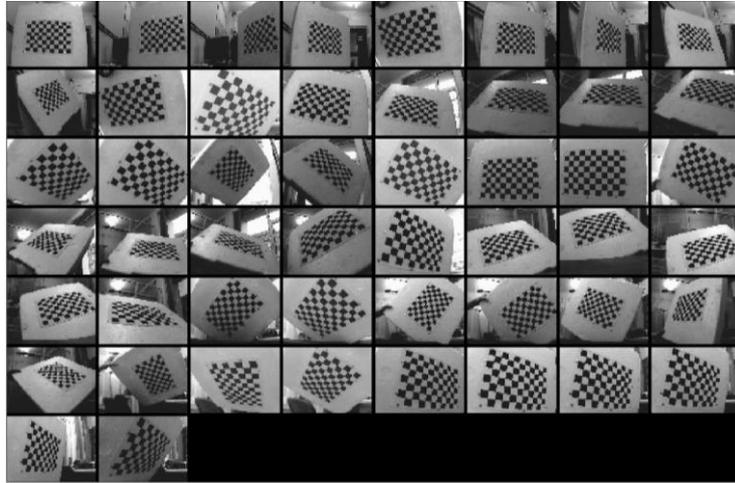


Figure 6. Image used for calibration

As the result of the calibration process using the Algorithm 2 the values of the intrinsic parameters are obtained:

- Principal point $P = (318.85122, 255.46648)$ (pixel)
- Focal length axis-x $f_x = 633.54607$ (pixel)
- Focal length axis-y $f_y = 634.02213$ (pixel)
- Skew $s = 0.0$
- Distortion coefficients $k_1 = -0.46378, k_2 = 0.28011, k_3 = 0.0, p_1 = 0.00083, p_2 = 0.00269$
- The total error is expressed in pixels $\sigma_x = 0.20879, \sigma_y = 0.24828$

It is necessary to keep in mind that only intrinsic parameters remain fixed because they are fixed values which depend on the camera assembly. On the other hand, the external parameters, rotation matrix and translation vector change as the BF or the camera move.

In the Figure 7 the extrinsic parameters by mean of the locations and orientations of the chessboard with respect to the CCS are shown which has been obtained during the calibration process.

- R_{CT} : Transformation matrix from the rotating table coordinate system $O_T X_T Y_T Z_T$ to the CCS.
- $R_{CW_{t_0}}, R_{CW_{t_1}}$: Transformation matrix at time t_0 and t_1 from the coordinate system $O_w X_w Y_w Z_w$ to the CCS obtained by the Algorithm 2.
- $T_{C_{t_0}}, T_{C_{t_1}}$: Transformation vectors at time t_0 and t_1 with respect to the camera coordinate system obtained by the Algorithm 2.
- T_W : Translation vectors with respect to the rotating table coordinate system $O_T X_T Y_T Z_T$.
- T_T : Translation vectors with respect to the CCS.

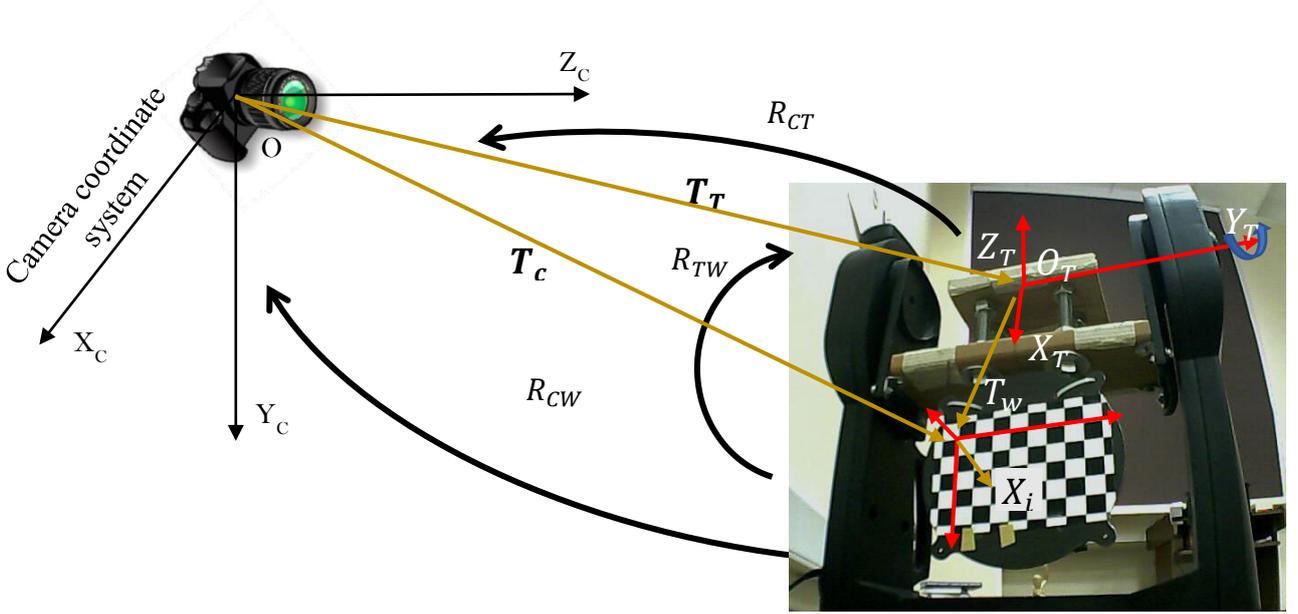


Figure 8. Testing schema using rotation table

From the equation (4.5) and (4.6) it is seen that

$$R_{CW_{t_0}} = R_{CT} R_{TW}$$

$$R_{CW_{t_1}} = R_{CT} R_{TW} R_{\alpha}$$

From the previous equations it is possible to obtain a direct formula to estimate the rotation matrix R_{α} (intrinsic rotation) with respect to $O_w X_w Y_w Z_w$, see the next equation

$$R_{\alpha} = R_{CW_{t_0}}^{-1} R_{CW_{t_1}} \quad (4.7)$$

The equation (4.7) can be rewritten using quaternions:

$$\Lambda_{\alpha} = \Lambda_{CW_{t_0}}^{-1} \circ \Lambda_{CW_{t_1}} \quad (4.8)$$

As it can be noticed in the previous equation, R_{α} depends on two consecutive rotations of coordinate system $O_w X_w Y_w Z_w$ which can be expressed as a function of

the unit quaternion Λ_{CW} . The rotation quaternion Λ_{CW} can be obtained by means of Algorithm 2 considering extrinsic parameters, orientation and the translation vector only. The intrinsic parameters, on the other hand, are not included in the parameters because they already have been determined during the calibration and they remain fixed.

It is important to mention that Λ_α represents the local rotation with respect to BF for period of time $\Delta t = t1 - t0$, and this rotation quaternion Λ_α is composed by the rotation angle α and its rotation axis \mathbf{n} as shown below:

$$\Lambda_\alpha = \begin{bmatrix} \lambda_{0\alpha} \\ \boldsymbol{\lambda}_\alpha \end{bmatrix} = \begin{bmatrix} \cos \frac{\alpha}{2} \\ \mathbf{n} \sin \frac{\alpha}{2} \end{bmatrix}. \quad (4.9)$$

From the previous equation α and \mathbf{n} are obtained by means of the next expressions

$$\begin{aligned} \alpha &= 2\arccos(\lambda_{0\alpha}), \\ \mathbf{n} &= \boldsymbol{\lambda}_\alpha \sin(\alpha/2) \end{aligned} \quad (4.10)$$

which allow the instantaneous angular velocity to be calculated from two consecutive rotation for period of time Δt as follows:

$$\mathbf{w}_{rel} = \frac{\alpha \mathbf{n}}{\Delta t} \quad (4.11)$$

where \mathbf{w}_{rel} represents the angular velocity of the rigid body with respect to the BF.

A testing with the rotating table which consists of three rotations of 1° , 2° and 3° around axis- Y_T is performed, then by using the equation (4.10) the measurement model accuracy is shown. In the Figure 9 it is noticed that the mean (μ) of consecutively rotations is very close to the true angle α with small standard deviation (σ).

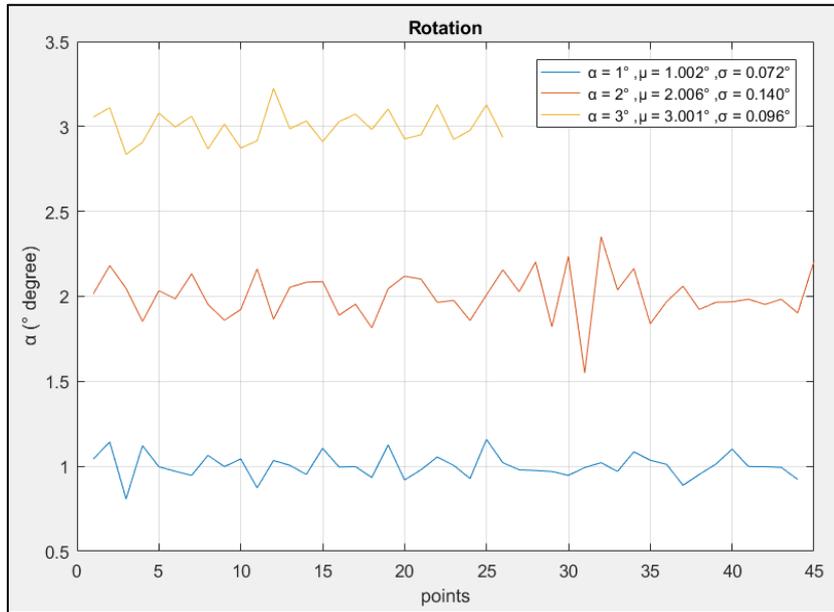


Figure 9. Detections for three rotations of 1° , 2° and 3° using the intrinsic parameters

Another testing is performed in order to know if it is possible to detect very small rotation angles such as 1 arcmin (0.0167°), 5 arcmin (0.0833°) and 15 arcmin (0.25°) using low resolution camera. As it can be seen in the Figure 10, the accuracy it is less as the rotation angle is smaller. On the other hand, the precision (σ) is still maintained.

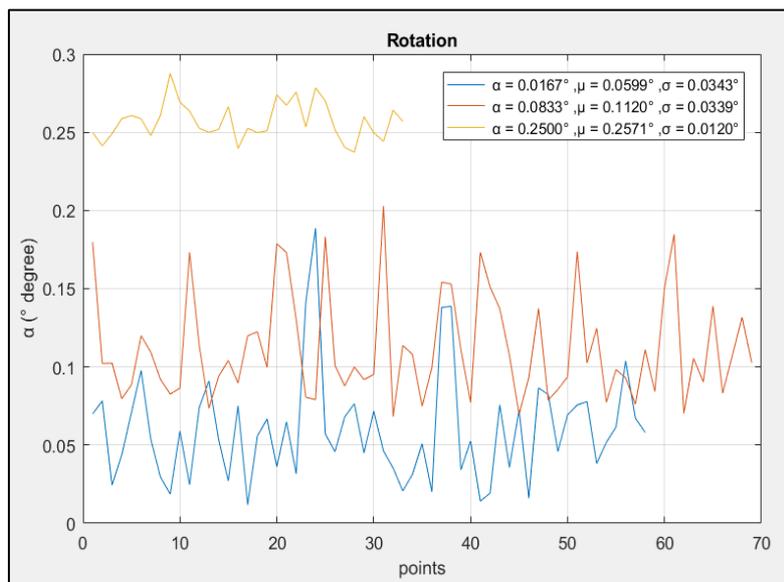


Figure 10. Detection of three rotations 1 arcmin (0.0167°), 5 arcmin (0.0833°) and 15 arcmin (0.25°) using the intrinsic parameters

Until this moment the testing has been performed using chessboard where a remarkable amount of points is provided. However, it is not possible to establish the correspondence between the point from the BF and the ICS automatically, this required the user support. It is very important that the program for image processing detects and localizes automatically and accurately the points of correspondence between the BF and the ICS, since the accuracy of the rotation matrix and translation vector depends on it.

It is shown in the Figure 11 that once the four points are detected and their position in the image is evaluated, it is impossible to determine which point is P1, P2, P3 or P4. Therefore, the correspondences are not possible to be determined.

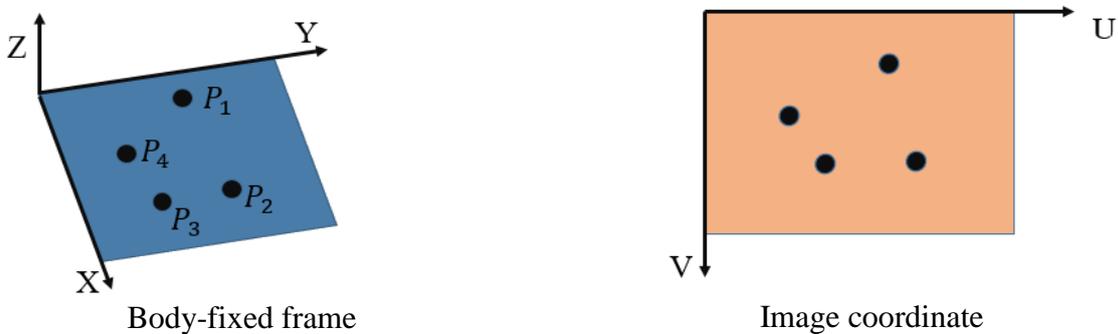


Figure 11. Example where correspondences are not possible determined

To solve this problem a pattern between each point can be used in order to determine the correspondences. In order to do that the utilization of the Aruco pattern is considered [15], [16]. It helps to establish the correspondence between the point from the BF and the ICS as it can be seen below.

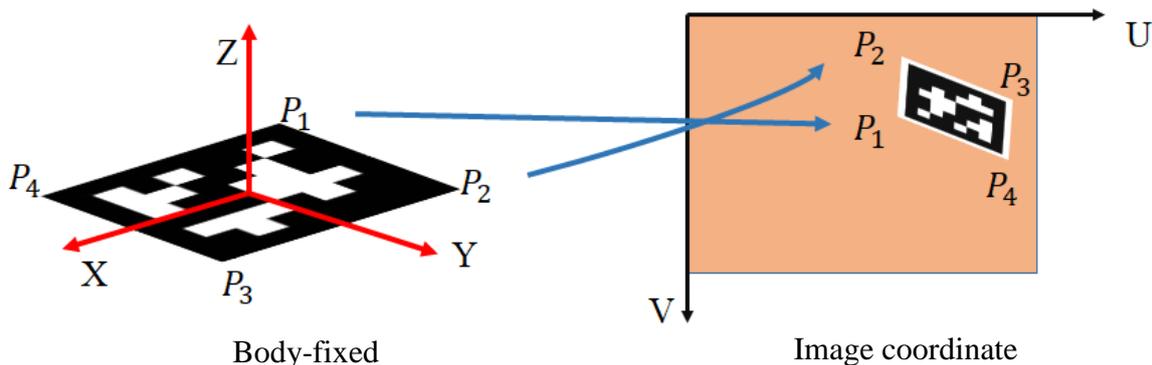


Figure 12. Correspondences determined by using Aruco patterns

In this experiment the correspondences are established automatically using the Aruco library. As it is understood, the measurement model's error is inherent, and in addition to that error, another source of errors appears such as: the error produced by change of brightness in the environment, by the digitization of the image, and by the algorithm for corner detection.

In the Figure 13 it is shown how the used Aruco pattern is installed on the rotating table

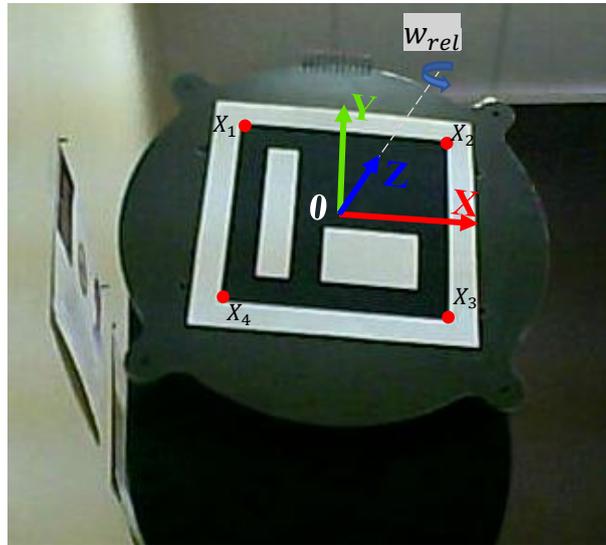


Figure 13. Aruco pattern and rotating table

In the Figure 14 it is shown how the location of a detected corner change for each image with the rotating table being static. The located corner present in coordinates $x(\text{pixel})$ and $y(\text{pixel})$ maximum standard deviation 0.11 and 0.12 respectively, the effects of this deviation are reflected in the precision of the rotation angle.

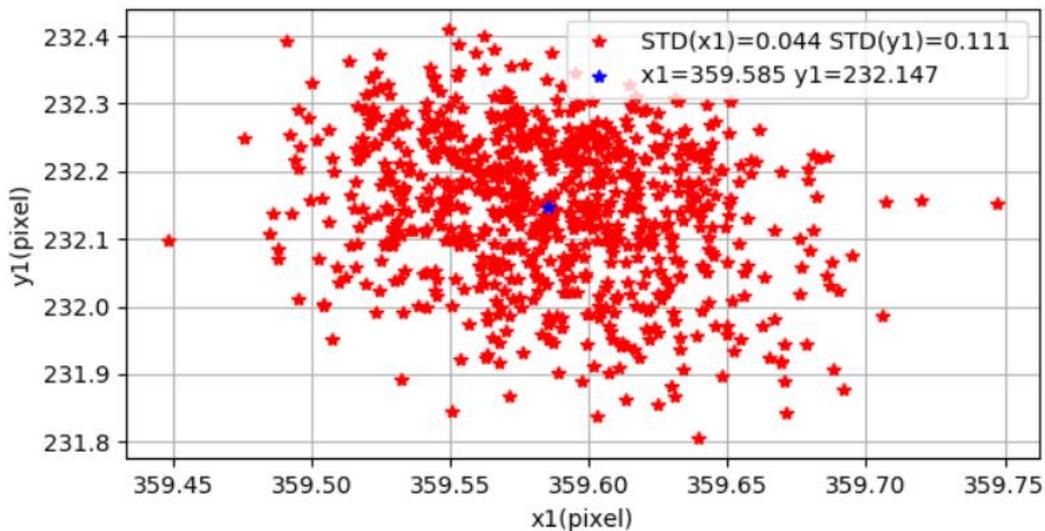


Figure 14. Mean and standard deviation (STD) of the point X1

In the Figure 15(a) it is shown that the rotation angle has a mean value of 158.77° , and the standard deviation (σ) equals to 0.156° . The distance showed in the Figure 15(b) represent the modulus of the translation vector T_C .

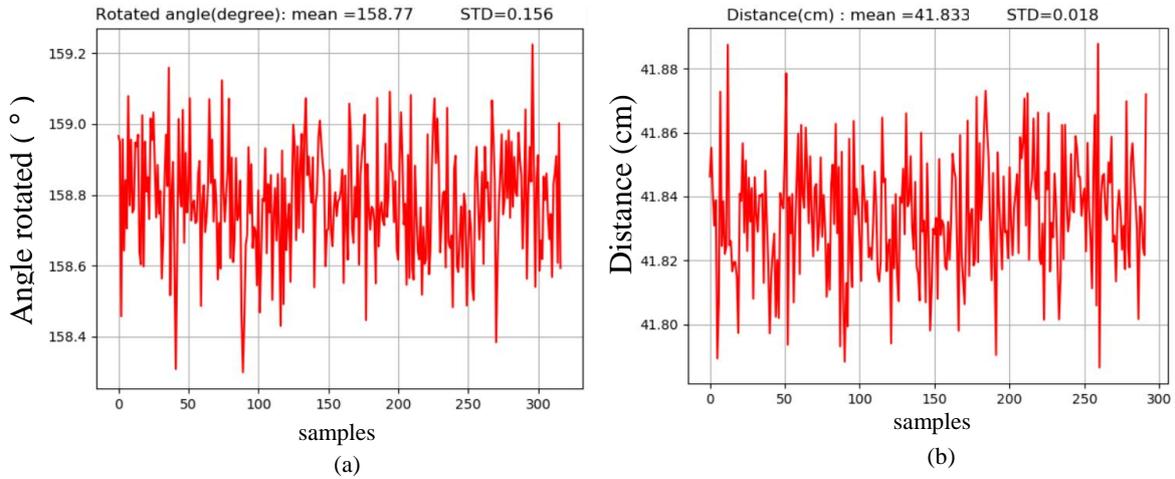


Figure 15. (a) Estimation of initial angle position with the rotating table being static. (b) Estimated distance with the rotating table being static.

Another experiment has been performed where the rotating table rotates 90° around the axis-Z. In the Figure 16, it is seen that, as it is expected, the estimated rotated angle is close to 90° . Additionally, the featuring of some peaks are seen, which appears due to the corner detector's errors.

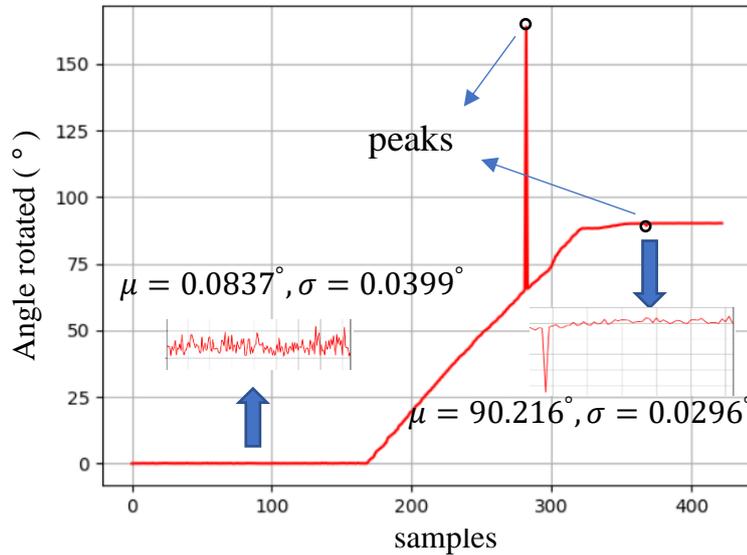


Figure 16. Estimated angle position with a rotation of 90°

The next experiment is focused on the local angular velocity calculation by using the camera FI8918W, previously calibrated in sub-section 4.4 while the rotating table rotates around the axis-Z.

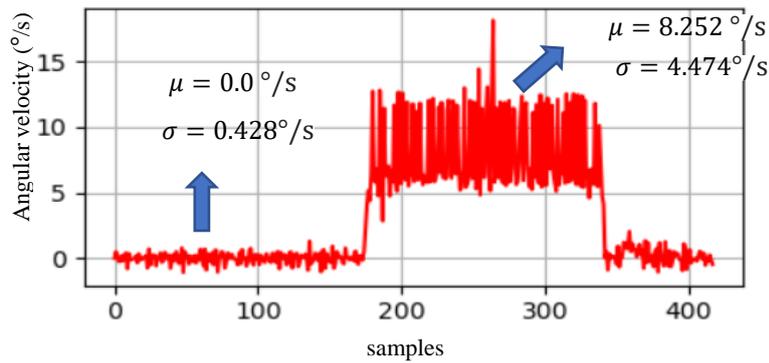


Figure 17. Angular velocity ($^{\circ}/s$)

The camera captures the object's movement by taking photographs every period of time Δt where $\Delta t = 1/15$ seconds, and at every object's movement sample the rotation quaternion is obtained by means of Algorithm 2, then by using the equation (4.11) the local angular velocity can be calculated as shown in the Figure 17.

As it can be observed in the Figure 17 the angular velocity calculation is strongly imprecise, its standard deviation σ can reach $4.47^{\circ}/s$, therefore to apply advanced technique for improving the angular velocity precision is to be recommended.

Conclusion

This work is dedicated to the problem of estimating the orientation of an object and its angular velocity by image processing. Two different approaches were considered: the orientation determination by means of the measurement model adapted for the use of quaternions, in addition the angular velocity calculation.

As result of quaternions use, the simulations showed that there is no difference with respect to the precision with its analog adapted measurement model for the Rodrigues rotation formula. However, using measurement model based on quaternions is a slight advantage in computing time.

Experiments for rotation matrix determination by means of quaternions showed high precision. However, the estimation of the angular velocity from consecutives rotation matrix has low precision. Thus, in order to improve the precision of the angular velocity measurement additional technique has to be implemented.

The good results obtained for orientation quaternions determination allow that this work can be easily integrated to another systems based on quaterions.

Reference

- [1] Hartley R., Zisserman A. Multi View Geometry in computer vision ,2nd ed, Cambridge University Press, 2004, 670c
- [2] Bouguet, J. Y. Camera Calibration Toolbox for Matlab, 2015, [Electronic resource], URL: http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed: 20.06.220).

- [3] Bradski G., Kaehler A. Learning OpenCV, 1st ed, O'Reilly Media, 2008, 555c
- [4] Orghidan R., Salvi J., Gordan M., Orza B. Camera calibration using two or three vanishing points // Proceedings of the Federated Conference on Computer Science and Information Systems, P. 123–130
- [5] Zhang Z., Flexible Camera Calibration By Viewing a Plane From Unknown Orientations // Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, Greece, September 20-27, 1999.
- [6] Tsai R. Y., A Versatile Camera Calibration Techniaue for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses // IEEE Journal on Robotics and Automation, 1987, Vol. 3, No. 4, P. 323 – 344.
- [7] Heikkilä J., Silvén O. A Four-step Camera Calibration Procedure with Implicit Image Correction // Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, USA, USA, June, 17-19 ,1997.
- [8] Dhome M., Richetin M., Lapreste J., Rives G. Determination of the Attitude of 3-D Objects from a Single Perspective View // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, Vol. 11, No. 12, P. 1265 – 1278.
- [9] Kim H., You B., Hager G. D., Oh S., Lee Ch. W. Three-Dimensional Pose Determination for a Humanoid Robot using Binocular Head System // Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), Kyongju, South Korea, South Korea, October, 17-21, 1999.
- [10] Zhong Z., Yi J., Zhao D. Pose Estimation and Structure Recovery from Point Pairs // Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, Spain, April, 18-22, 2005
- [11] Zhang X., Zhang Z., Li Y., Zhu X. Robust camera pose estimation from unknown or known line correspondences // OSA Journal on Applied Optics,2012, Vol. 51, No. 7, P. 936-948.
- [12] Gallego G., Scaramuzza D. Accurate Angular Velocity Estimation with an Event Camera // IEEE Robotics and Automation Letters, 2017, Vol. 2, No. 2, P. 632 - 639.
- [13] Wang S., Li Q., Guan B. A computer vision method for measuring angular velocity // Optics and Lasers in Engineering, Vol. 45, No. 11, P. 1037-1048.
- [14] Zhang Y, Wen C, Zhang Y. Estimation of motion parameters from blurred images // Pattern Recognition Letters, 2000, Vol. 21, No. 5, P. 425-433.
- [15] Romero-Ramirez F. J., Muñoz-Salinas R., Medina-Carnicer R. Speeded up detection of squared fiducial markers // Image and Vision Computing, 2018, vol 76, P 38-47.
- [16] Garrido-Jurado S., Muñoz Salinas R., Madrid-Cuevas F.J., Medina-Carnicer R. Generation of fiducial marker dictionaries using mixed integer linear programming // Pattern Recognition, 2016, Vol. 51, P. 481-491.
- [17] Boguslavsky A.A., Sazonov V.V., Sokolov S. M., Smirnov A. I., Saigirae

Kh.S. Automatic vision-based monitoring of the spacecraft docking approach with the International Space Station // Proceedings of the First International Conference on Informatics in Control, Automation and Robotics, 2004, P. 79-86

- [18] Ivanov D.S., Karpenko S.O., Ovchinnikov M.Yu., Sakovich M.A. Satellite relative motion determination during separation by video processing, Keldysh Institute preprints, 2012, № 57, 24 p.
- [19] Koptev M.D, Proshunin N.N., Ivanov D.S., Motion determination of microsatellite control system mock-ups on aerodynamic test-bed by using monocular vision, Keldysh Institute preprints, 2015, № 109, 32 c.

Content

Introduction	3
1. Problem statement	4
2. Measurement model.....	7
2.1. Transformation from BF to CCS	7
2.2. Projection of the points from CCS into the image plane.....	7
2.3. Lens distortion	8
2.4. Transformation from sensor plane to ICS	8
2.5. Measurement model based on Quaternions.....	9
2.6. Measurement model gradient	10
3. Linearization of the measurement model.....	13
4. Calibration Algorithm.....	16
4.1. Initialization of parameters	16
4.2. Optimization process	17
4.3. Algorithm for camera calibration	21
4.4. Application algorithm for camera calibration	22
4.5. Testing measurement model.....	23
Conclusion.....	30
Reference.....	30
Content	32