Exhaustive Simulation Approach for a Virtual Camera Calibration Evaluation in Gazebo

2022 8th International Conference on Automation, Robotics and Applications (ICARA) | 978-1-6654-8383-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICARA55094.2022.9738581

Tatyana Tsoy
Intelligent Robotics Dept.
Kazan Federal University
Kazan, Russia
tt@it.kfu.ru

Ramil Safin
Intelligent Robotics Dept.
Kazan Federal University
Kazan, Russia
safin.ramil@it.kfu.ru

Edgar A. Martinez-Garcia
Industrial Engineering & Manufacturing Dept.
The Autonomous University of Ciudad Juarez
Ciudad Juarez, Mexico
edmartin@uacj.mx

Sumantra Dutta Roy
Electrical Engineering Dept.
Indian Institute of Technology Delhi
New Delhi, India
sumantra@ee.iitd.ac.in

Subir Kumar Saha
Mechanical Engineering Dept.
Indian Institute of Technology Delhi
New Delhi, India
saha@mech.iitd.ac.in

Evgeni Magid
Intelligent Robotics Dept.
Kazan Federal University
Kazan, Russia
magid@it.kfu.ru

***Abstract*—Camera calibration is an essential research field with a high potential for emerging algorithms and calibration targets. In this work, we present a virtual experiments approach in Gazebo simulator for exhaustive camera calibration methods and calibration targets evaluation. Key steps of the camera calibration workflow were adapted to the Gazebo simulation approach. We proposed a virtual camera calibration evaluation pipeline that includes camera modeling and calibration target's pose generation in a viewing frustum. Experiments in the Gazebo demonstrated virtual environment feasibility for camera calibration evaluation while comparing checkerboard and circle grid targets, and allowed to achieve an acceleration of more than 30 times compared to the real-time. Experimental results exhibited a need for additional calibration steps incorporation, such as outliers rejection and optimal calibration target poses generation.**

Keywords—camera calibration, photogrammetry, computer vision, automation, simulation, ROS Framework, Gazebo.

I. Introduction

In the field of machine vision, information about 3D objects is obtained from digital images. There are multiple steps in an image formation pipeline, most of which degrade an accuracy of resulting measurements represented by images. Thus, to retrieve a consistent and reliable metric information about an environment, a camera calibration procedure is required.

Multiple applications require camera calibration [1], including visual navigation, medicine (e.g., medical surgery and saturation), 3D scene reconstruction and structure from motion, camera and object localization, simultaneous localization and mapping (SLAM, [2]), etc. Camera calibration estimates a model of a 3D to 2D mapping by finding correspondences between points in the world and image pixels. A relationship between a point in a 3D space and its corresponding image point is determined by a camera model. Generally, camera calibration computes camera parameters by observing a particular calibration target.

Relevance of the camera calibration problem arises due to the existence of a variety of calibration approaches and objects suitable for particular conditions. Such conditions may include lack of illumination, severe weather conditions, partial occlusion (objects blocking a calibration target visibility), inherent hardware noise, etc.

Most of the time, camera calibration is executed manually in a laboratory environment. However, there are some inevitable disadvantages including time consumption in exhaustive large-scale experiments, an inability to control and monitor environmental conditions, and a complicated selection of appropriate hardware configurations [3]. On the other hand, virtual environments facilitate experiments reproduction in different conditions. There is no need for an elaborate camera calibration setup anymore, while hardware capabilities enable parallel computations. All together, these reduce typical to real-world scenarios heavy consumption of a valuable time resource.

This work presents a virtual experiments approach in the Gazebo simulator for an exhaustive evaluation of traditional camera calibration algorithms and calibration targets. We demonstrated our approach feasibility by comparing two planar calibration patterns, checkerboard and circle grid, in various scenarios. The scenarios included camera calibration in the presence of different levels of Gaussian noise $N(0, \sigma^2)$ and a varying number of control points. An acceleration of more than 30 times was achieved in reference to the real-time.

II. Related Work

A. Overview

Camera calibration is a process of calculating parameters of a mathematical model of a camera by performing experiments using calibration targets [4]–[7] or natural features of an environment [8]–[11]. The most popular method is to calibrate a camera using images of a calibration target.

Traditionally, the camera calibration procedure can be divided into the following major stages:

1. ***Modeling*** – an approximation of physical and optical characteristics of a camera and a calibration target;
2. ***Dataset collection*** – an accumulation of calibration samples by viewing the calibration target at various poses relatively to the camera;
3. ***Calibration*** – analytical or iterative estimation of the camera parameters using the calibration target's features.

Camera parameters are represented by a camera model. The most prominent one is a perspective model, which is based on pin-hole principles [12]. Images are formed by means of 3D points perspective projection onto an image plane; light passes through an optical center and forms a pixel on the image (Fig. 1). The camera model parameters are divided into two groups:

- ***Internal*** – simulate internal geometry and optical characteristics of an image sensor (e.g., a focal length, an optical center, a pixel skew coefficient) and a lens (e.g., a distortion); those parameters determine how the light passes through the lens onto an image plane;
- ***External*** – define a position and an orientation of a calibration target relative to a camera.

There exist linear and non-linear calibration approaches. Linear techniques estimate camera parameters analytically. On the other hand, the real world and the camera apparatus contain uncertainties. Most of the time researchers are concerned about non-linearities such as noise and lens deformation [13].

Non-linear optimization techniques estimate camera parameters iteratively by minimizing a certain cost function [14]. Usually, the cost function is defined as a distance between image points and their modeled projections, which is referred as a reprojection error. Two-step approaches combine benefits of both methods by computing an initial guess analytically and further using it in non-linear optimization [15].

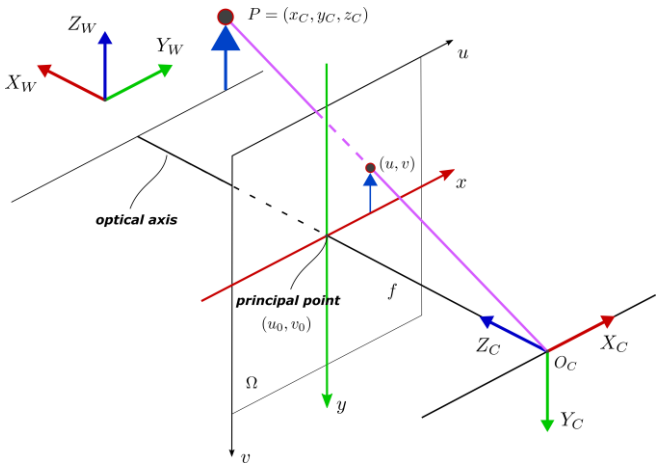


Fig. 1. Pin-hole camera model: perspective projection of a 3D world point $P = (x_c, y_c, z_c)$ onto a 2D image plane Ω. Light from P passes through the optical center O_C to form a pixel at (u, v). Adapted from the OpenCV documentation (Copyright © 2021 Itseez).

B. Seminal Works

One of the first works on camera calibration was done by Hall et al. in 1982 [16]. It considered an implicit camera calibration approach that estimated a 3×4 camera transformation matrix. Later, in 1986, Faugeras proposed to extract explicit camera parameters from the camera transformation matrix [17]. Non-linear optimization explicit camera calibration approaches incorporated lens distortion into a camera model; some works considered only a radial distortion [18] while others included up to three types of distortion [15].

Tsai was one of the first to propose a 2-step method composed of an analytical initial guess and a further non-linear optimization [18]. This method utilized only one image of a 3D checkerboard calibration target. Heikkila and Silven in their work presented a 4-step approach as an extension of Tsai's method [19]. They incorporated more distortion terms and considered circle control points.

Later, Sturm and Maybank proposed a camera calibration algorithm based on multiple planar objects and analyzed its singularities [20]. The prominent work of Zhang made it easier for users to accomplish a camera calibration procedure by using a single checkerboard planar pattern that could be printed on paper [4]. Nowadays, there exist a number of camera calibration algorithms that are based on Zhang's work, including popular implementations in OpenCV and Matlab.

C. Calibration Templates

A camera calibration target is a 1D, 2D, or 3D object with a predefined geometry. One of the most widespread calibration targets is Zhang's planar checkerboard [4] (Fig. 2a). The idea behind the pattern is to create strong edge gradients that can be detected with high accuracy at a sub-pixel level.

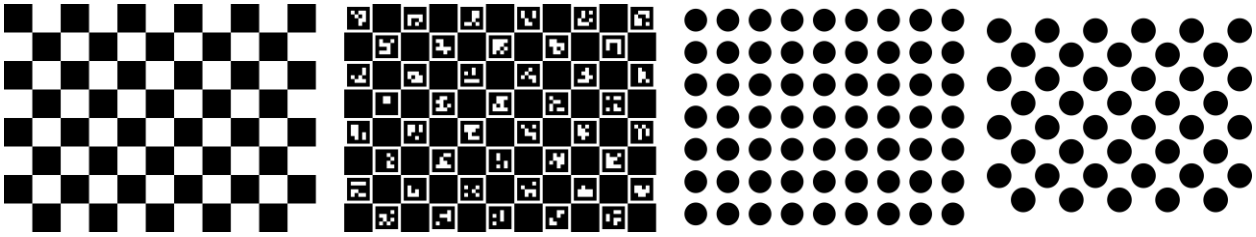
A circle grid is another representative of planar calibration targets [21]. It consists of circle blobs placed on a grid (Fig. 2c). In comparison to a checkerboard pattern, a circle grid is considered to be less precise due to inflicted perspective distortion effects [22].

State-of-the-art calibration targets boost calibration accuracy and robustness even further by embedding auxiliary structural elements such as fiducial markers [3], [23]. One of the examples is a ChArUco board [6] (Fig. 2b) – ArUco tags are embedded into checkerboard squares. Fiducial markers facilitate control points detection by interpolating their position.

III. Gazebo Virtual Environment

In this section, we present a Gazebo simulation environment that was constructed for exhaustive traditional camera calibration algorithms and calibration targets evaluation. The Gazebo is an open-source simulator licensed under Apache 2.0. Model spawning API enables users to create various robots and other physical objects in the existing Gazebo world. It supports several physics engines and has 3D graphics rendering capabilities [24], [25].

Perception of a virtual world is possible due to sensor plugins. Out of the box, the Gazebo provides a laser range finder, monocular and stereo cameras, wide-angle and depth cameras, an inertial measurement unit, contact, and force-torque sensors.



(a) (b) (c) (d)

Fig. 2. Planar calibration targets (templates): (a) checkerboard (b) ChArUco board (c) circle grid (d) asymmetric circles.

Our simulated environment contains a stationary monocular camera and a moving calibration target (Fig. 3). The camera is placed over a rectangular stand of a 1 m height. The calibration target is a planar pattern with precisely known geometrical properties. Six sources of a uniform directed light are placed at different sides of the scene and a constant ambient light is enabled throughout the 3D scene. Shadows are disabled to avoid calibration target detection issues caused by shadow gradients.

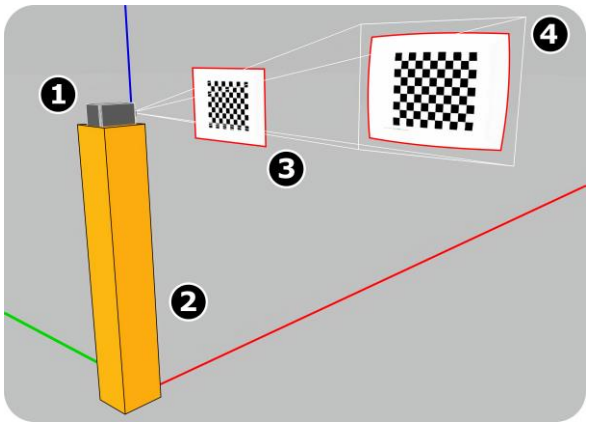


Fig. 3. Gazebo simulator camera calibration evaluation virtual world setup. The camera (1) is placed on the orange stand (2) looking at the planar calibration target (3). The rendered image (4) demonstrates a medium radial distortion presence.

Gazebo monocular camera plugin emulates a pin-hole camera model. Lens optical deformation is represented by radial and tangential distortion components [13]. It is possible to incorporate normally distributed noise $N(\mu, \sigma^2)$ into measurements, i.e., images. Other adjustable imaging parameters include a focal length, an image resolution, a field of view, a frame rate, and a color space. The calibration target is a white planar surface with a particular 2D visual material applied (i.e., plotted) on the surface, e.g., a checkerboard or circle grid pattern. Material rendering in Gazebo is handled by the Blinn–Phong reflection model.

IV. Calibration Routine

There exist several projects that use Gazebo for virtual camera calibration, e.g., *calibration_gazebo* ROS package[1]. However, those implementations pursue manual calibration as opposed to automatic approach. We define the following camera calibration simulation steps:

1. **Camera modelling**. Creating a simplified mathematical camera model in Matlab to analyze its 3D viewing frustum.
2. **Calibration targets poses generation**. Sampling good 6D poses from the viewing frustum for a calibration target relatively to the camera.
3. **Calibration procedure**. Building blocks of the calibration process – calibration target movement control, image capture, simulation configuration management, etc.

Next, we provide more details about each of these steps.

A. Camera Modeling

Traditional camera calibration involves moving a particular calibration target (pattern) within a 3D scene in a way that the pattern is partially or fully visible to a camera. A viewing frustum is a part of the scene visible to the camera. It represents a pyramid with an apex at the camera's optical center sliced from a top and a bottom by near and far planes, respectively (Fig. 4).

We model the viewing frustum for given camera parameters with horizontal field of view $hfov$, aspect ratio as_ratio, camera's closest $clip\_close$ and farthest $clip\_far$ working (viewing) distances. Calculations use simple geometry and trigonometry rules of triangles. For example, computation of a width and a height of the frustum's rectangular section S at distance $view\_dist$ along the camera's optical axis is performed using the following equations:

$$S_{width} = 2 \cdot \tan\left(\frac{hfov}{2}\right) \cdot view\_dist$$
$$S_{height} = S_{width} \cdot as\_ratio^{-1} \quad (1)$$

B. Calibration Target Poses Generation

One of the most important parts of a traditional camera calibration is the camera movement relatively to a calibration target (or vice-versa). Most of the existing algorithms require users to capture images of a calibration target on their own. Relying solely on a user experience might lead to calibration errors due to motion blur and incorrect or even singular poses in the camera-target system. However, a number of works on interactive calibration approaches suggest scoring poses based on heuristics, error minimization in pixels domain, and singular poses detection [7], [26], [27].

[1] https://github.com/oKermorgant/calibration_gazebo

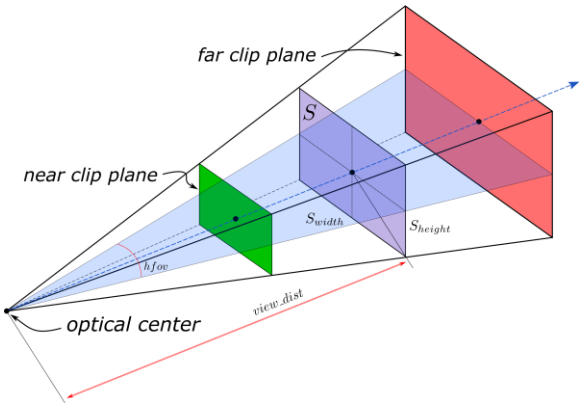


Fig. 4. Modeled camera viewing frustum.

Our approach generates 6D poses of a pattern inside a camera's viewing frustum. We employ the following rules in order to achieve better calibration results, both in accuracy and robustness:

a. Keep the target close to a real working distance – poses are scored in a working distance's vicinity.

b. Maximize an angular spread of a calibration target's orientation.

c. Cover the entire viewing frustum. Sample more poses close to distortion edges.

d. Avoid poses where the target is not parallel to any of camera's optical axes (a pin-hole singularity).

The pose generation workflow is demonstrated in Fig. 5. We start from generating unique random combinations of roll-pitch-yaw angles (RPY) with respect to the camera's optical frame within $[\theta_{min}, \theta_{max}]$ degrees, where $3 \leq \theta_{min} \leq \theta_{max} \leq 45$. Next, 50% of the angles are negated to achieve the angle spread within $[\theta_{min}, \theta_{max}] \cup [-\theta_{max}, -\theta_{min}]$.

Next, a set of viewing distances D is generated from a uniform distribution. They are bound by the minimal D_{min} ($D_{min} \geq 0$) and maximal D_{max} ($D_{max} \geq D_{min}$) values in meters. After that, a viewing frustum plane VF is computed at each generated viewing distance (see subsection IV-A). To avoid cases where the calibration target is outside the frustum at particular orientations, a configuration space VF_{CS} is estimated.

Finally, we sample calibration target positions Pos inside the viewing frustum plane's C-space (see Fig. 5, step 6). Given a set of calibration target's angles and positions with respect to the camera's optical frame, we form the resulting 6D poses ($Poses$). The poses are stored on a disk in the comma-separated values (CSV) file format.

C. Calibration Procedure

The calibration procedure takes place in the Gazebo virtual environment. The camera is fixed, and the calibration target moves between generated 6D poses using Gazebo control API. Particularly, we integrate with ROS Framework to send controls and obtain information about the virtual world. In order to drastically decrease the simulation time (up to 30 times of the real-time), *real_time_update_rate* is set to 30 000.

Poses are loaded from the file system into the program's memory. We change a current calibration target's pose using */set_model_state* ROS service. When the target successfully moves to a desired pose, we capture images from the camera. The stream of images is provided through */image_raw* ROS topic. Calibration samples are collected and stored in a separate directory for further offline calibration and evaluation.

The calibration target's material (i.e., a calibration pattern) could be dynamically replaced by updating the corresponding Collada file (with **.dae* extension). For this purpose, a set of calibration pattern images were generated in advance.

V. Experiments

A. System Setup

Virtual experiments were conducted on a computer with the following characteristics:

- Intel Core i7-8750H CPU (6 cores @ 2.2 GHz)
- 16 GB RAM
- 512 GB SSD
- GTX 1050 Ti GPU (4GB GDDR5)
- Ubuntu 20.04 LTS OS, ROS Melodic, Gazebo 11

B. Methodology

In our simulations, we employed the pin-hole camera model, which incorporates three radial and two tangential distortion components. Modeled camera properties are the following:

- Frame rate: 20 fps
- Color format: 8-bit Grayscale
- Horizontal field of view: 60 degrees
- Image resolution: 640 × 480 (4:3 aspect ratio)
- Viewing frustum: a near clipping (0.1 m) and a far clipping plane (5 m)

All calibration targets were embedded into a 2D object of dimensions specified by the A4 paper standard of 297 mm × 210 mm. In the experiments we employed two calibration patterns:

1. Checkerboard with each square side length of 20 mm
2. Circle grid with each circle diameter of 15 mm and inter-circle gaps of 21 mm

The following experiment scenarios were performed:

a. **Control points**. Varying a number of calibration target's control points: 5 × 7 and 7 × 9.

b. **Noise**. Incorporating an independent noise $X \sim N(0, \sigma^2)$ for each pixel of an image: *none* and $\sigma = 0.001$.

c. **Patterns**. Calibration target's patterns: checkerboard and circle grid (Fig. 2a and Fig. 2c).

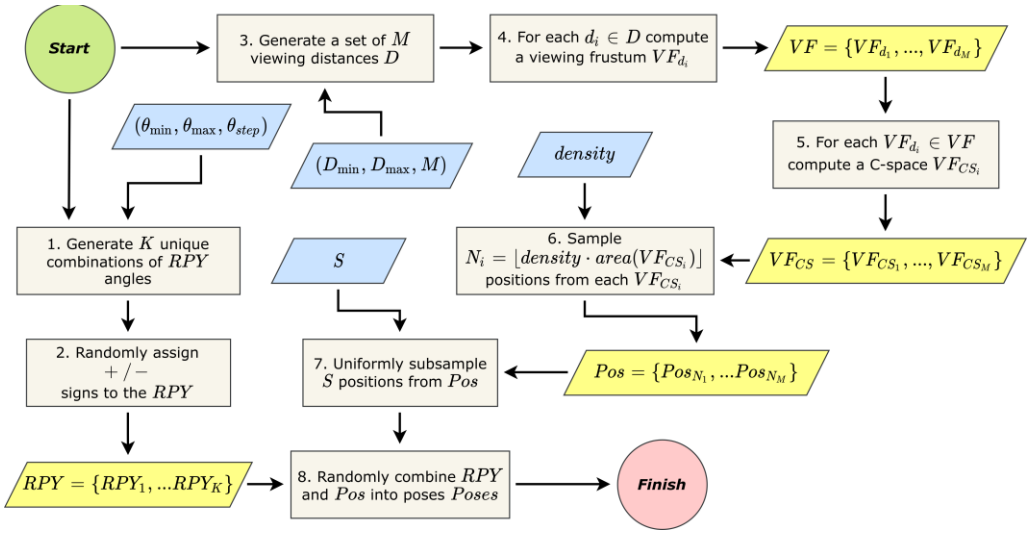


Fig. 5. 6D calibration target poses generation flowchart. Yellow color is for computed variables, blue is for input arguments. Arrows show the execution flow.

TABLE I. RESULTS OF THE CAMERA CALIBRATION EVALUATION IN THE VIRTUAL ENVIRONMENT

Target	Control Points	Noise (*stddev*)	Samples' RMS reprojection error [px]						
			25	50	100	150	200	250	300
Checkerboard	5 × 7	*none*	0.093 ± 0.005	0.095 ± 0.005	0.093 ± 0.003	0.095 ± 0.003	0.094 ± 0.002	0.095 ± 0.002	0.093 ± 0.001
		0.001	0.101 ± 0.004	0.105 ± 0.004	0.104 ± 0.004	0.104 ± 0.004	0.106 ± 0.003	0.105 ± 0.001	0.103 ± 0.001
	7 × 9	*none*	0.101 ± 0.004	0.102 ± 0.004	0.099 ± 0.003	0.103 ± 0.004	0.101 ± 0.002	0.101 ± 0.002	0.099 ± 0.001
		0.001	0.109 ± 0.003	0.111 ± 0.003	0.109 ± 0.003	0.11 ± 0.002	0.112 ± 0.002	0.111 ± 0.001	0.11 ± 0.001
Circle grid	5 × 7	*none*	1.243 ± 0.168	1.24 ± 0.149	1.22 ± 0.085	1.36 ± 0.113	1.25 ± 0.076	1.205 ± 0.053	1.26 ± 0.078
		0.001	1.261 ± 0.146	1.269 ± 0.176	1.193 ± 0.092	1.359 ± 0.107	1.257 ± 0.08	1.191 ± 0.047	1.315 ± 0.042
	7 × 9	*none*	1.19 ± 0.129	1.176 ± 0.124	1.139 ± 0.065	1.237 ± 0.086	1.185 ± 0.094	1.141 ± 0.024	1.2 ± 0.067
		0.001	1.174 ± 0.112	1.201 ± 0.12	1.125 ± 0.067	1.25 ± 0.095	1.19 ± 0.087	1.141 ± 0.035	1.195 ± 0.06

For each scenario, we used 10 sets of $S = \{25, 50, 100, 150, 200, 250, 300\}$ calibration samples, i.e., calibration target 6D poses. Samples viewing distance lay within $D_{min} = 0.45$ m and $D_{max} = 0.7$ m. Poses were sampled at different viewing distances according to the $density = 200$ parameter that was measured in samples per square meter. The angle spread was set to $\theta \in [-45, 45]$ degrees.

The calibration algorithms, which are a part of the OpenCV library, were used as-is without calibration parameters modifications. In our evaluation, we employed a root-mean-square (RMS) reprojection error (RE) measured in pixels, i.e., the error of a 3D point projection onto the image plane using estimated camera parameters. We computed a mean and standard deviation for the calibration sample sets.

VI. RESULTS

Experimental results are shown in Tab. I. RMS RE was computed for 5 × 7 and 7 × 9 the checkerboard and the circle grid calibration boards. Measurements were grouped by the calibration target, the number of control points, and the presence of Gaussian noise.

One can notice, that there is no significant improvement in accuracy when the number of calibration samples is increased. It could be caused by the presence of outliers. RMS RE reflects how the estimated camera parameters fit our calibration samples. However, a case when RMS RE is close to zero, but estimated camera parameters are far behind real camera parameters is possible (i.e., overfitting).

Overall, the checkerboard demonstrated a significant superiority compared to the circle grid. This might be explained by a lack of a blob detector customization. It requires fine-tuning for obtaining accurate circle center positions. Moreover, circles detection is sensitive to a perspective distortion. As it was mentioned in subsection V-B, we maximized the angle spread in our generated calibration target poses, so the perspective distortion is possible.

Introducing a low level of noise ($\sigma = 0.001$) slightly decreased the performance of both calibration templates. Yet, going from 5 × 7 to 7 × 9 control points did not influence the results. We believe that in order to notice the effects one needs to increase the number of control points even further.

VII. CONCLUSIONS

Camera calibration is an established part of computer vision and photogrammetry research areas. There exists a large number of algorithms that make use of different calibration targets and mathematical background. However, even up to this day, new calibration approaches and calibration targets unveil new prospects into high-accuracy and robust camera calibration.

The main difficulty arises from numerous parameters that influence calibration results, including a pattern selection, a

number of required (by the pattern) control points, an ideal working distance, robustness to noise, and arbitrary occlusions. Typically, calibration methods are evaluated in laboratory environments and the calibration procedure requires significant time, an elaborate setup of the environment, illumination, and other physical conditions control.

In this work, we demonstrated the feasibility of using virtual environments for camera calibration algorithms evaluation in different scenarios. Acceleration of up to 30 times compared to the real-time was achieved, which enables one to conduct a large number of experiments with different calibration algorithms and targets in various conditions. A virtual camera calibration evaluation pipeline was proposed. It includes camera modeling and calibration target's pose generation in a viewing frustum. The calibration target's poses were refined to avoid singularities.

Experimental results exhibited a need for incorporating additional calibration steps, such as outliers rejection and optimal calibration target poses generation. The expected accuracy improvement with an increasing number of samples and control points was not observed.

ACKNOWLEDGMENT

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program (PRIORITY-2030). The first author acknowledges the support of RFBR, project number 20-38-90257.

REFERENCES

[1] Shabalina, K., Sagitov A., Svinin, M. and Magid E. "Comparing fiducial markers performance for a task of a humanoid robot self-calibration of manipulators: A pilot experimental study." In International Conference on Interactive Collaborative Robotics, pp. 249-258. Springer, Cham, 2018.

[2] Mingachev, E., Lavrenov, R., Tsoy, T., Matsuno, F., Svinin, M., Suthakorn, J., and Magid, E. "Comparison of ros-based monocular visual slam methods: Dso, ldso, orb-slam2 and dynaslam." In International Conference on Interactive Collaborative Robotics, pp. 222-233. Springer, Cham, 2020.

[3] A. Zakiev, K. Shabalina, T. Tsoy, and E. Magid, "Pilot virtual experiments on aruco and artag systems comparison for fiducial marker rotation resistance," in Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings". Springer, 2020, pp. 455–464.

[4] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.

[5] F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, no. 5, pp. 266–272, 2006.

[6] G. H. An, S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang, "Charuco board-based omnidirectional camera calibration method," Electronics, vol. 7, no. 12, p. 421, 2018.

[7] A. Richardson, J. Strom, and E. Olson, "Aprilcal: Assisted and repeat-able camera calibration," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013, pp. 1814–1821.

[8] R. Cipolla, T. Drummond, and D. P. Robertson, "Camera calibration from vanishing points in image of architectural scenes." in BMVC, vol. 99. Citeseer, 1999, pp. 382–391.

[9] C. S. Fraser, "Digital camera self-calibration," ISPRS Journal of Photogrammetry and Remote sensing, vol. 52, no. 4, pp. 149–159, 1997.

[10] E. E. Hemayed, "A survey of camera self-calibration," in Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003. IEEE, 2003, pp. 351–357.

[11] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in European conference on com-puter vision. Springer, 1992, pp. 321–334.

[12] B. Poling, "A tutorial on camera models," University of Minnesota, pp. 1–10, 2015.

[13] J. G. Fryer and D. C. Brown, "Lens distortion for close-range photogrammetry," Photogrammetric engineering and remote sensing, vol. 52, pp. 51–58, 1986.

[14] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in International workshop on vision algorithms. Springer, 1999, pp. 298–372.

[15] J. Weng, P. Cohen, M. Herniou et al., "Camera calibration with distortion models and accuracy evaluation," IEEE Transactions on pattern analysis and machine intelligence, vol. 14, no. 10, pp. 965–980, 1992.

[16] E. L. Hall, J. B. Tio, C. A. McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision," Computer, vol. 15, no. 12, pp. 42–54, 1982.

[17] O. D. Faugeras, "The calibration problem for stereo," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1986, pp. 15–20.

[18] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," IEEE Journal on Robotics and Automation, vol. 3, no. 4, pp. 323–344, 1987.

[19] J. Heikkila and O. Silven,´ "A four-step camera calibration procedure with implicit image correction," in Proceedings of IEEE computer society conference on computer vision and pattern recognition. IEEE, 1997, pp. 1106–1112.

[20] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 1. IEEE, 1999, pp. 432–437.

[21] P.-F. Luo and J. Wu, "Easy calibration technique for stereo vision using a circle grid," Optical Engineering, vol. 47, no. 3, p. 033607, 2008.

[22] J. Mallon and P. F. Whelan, "Which pattern? biasing aspects of planar calibration patterns and detection methods," Pattern recognition letters, vol. 28, no. 8, pp. 921–930, 2007.

[23] T. Tsoy, A. Zakiev, K. Shabalina, R. Safin, E. Magid, and S. K. Saha, "Validation of fiducial marker systems performance with rescue robot servosila engineer onboard camera in laboratory environment," in 2019 12th International Conference on Developments in eSystems Engineering (DeSE). IEEE, 2019, pp. 495–499.

[24] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3. IEEE, 2004, pp. 2149–2154.

[25] B. Abbyasov, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Svinin, and Magid, "Automatic tool for gazebo world construction: from a grayscale image to a 3d solid model," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 7226–7232.

[26] P. Rojtberg and A. Kuijper, "Efficient pose selection for interactive camera calibration," in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2018, pp. 31–36.

[27] W. Lei, M. Xu, F. Hou, W. Jiang, C. Wang, Y. Zhao, T. Xu, Y. Li, Zhao, and W. Li, "Calibration venus: An interactive camera calibration method based on search algorithm and pose decomposition," Electronics, vol. 9, no. 12, p. 2170, 2020.