Multi-body Self-Calibration: Supplementary Material

Andrea Porfiri Dal Cin
andrea.porfiridalcin@polimi.it
Giacomo Boracchi
giacomo.boracchi@polimi.it
Luca Magri
luca.magri@polimi.it
Politecnico di Milano

1 Introduction

This document provides additional materials omitted from the main manuscript due to space restrictions. In Section 2, we describe the proposed multi-body `Amiibo` dataset, and the camera calibration process used to obtain ground-truth camera internal parameters. In Section 3, we integrate the experimental validation presented in the main manuscript with additional results related to the performance of the proposed method. Specifically, we present the results of our pipeline on synthetic datasets, discuss the impact of critical motions, and comment on the computational burden of the proposed approach. The Supplementary Material includes a Zip file containing a MATLAB implementation of the proposed method and several running examples to test it on both synthetic and real datasets. The code is available at [1].

2 Amiibo Dataset

In this section, we describe the proposed `Amiibo` dataset introduced in the main manuscript and provide additional details on the camera calibration process. Introducing a new dataset stems from the scarcity of public, *calibrated*, multi-body datasets. The `Hopkins155` dataset provides several multi-body sequences with diverse motions, but it only contains short video sequences, and the background is always visible and often dominant w.r.t. the moving objects. For a more in-depth evaluation, we propose the `Amiibo` dataset, which contains challenging scenes in which the background is not always visible or in focus. In addition, images are taken from different viewpoints, and rigid objects exhibit complex shapes and a significant variation in size w.r.t. the camera field of view. The dataset and calibration parameters are available at [1].

Calibration The images were captured with a Canon EOS 400D. The calibration was obtained with the MATLAB R2021b Camera Calibration app included in the Computer Vision

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

(a) S1-Amiibo - IMG_4256

(b) S2-Amiibo - IMG_4282

(c) S3-Amiibo - IMG_4290

(d) M1-Amiibo - IMG_4208

(e) M2-Amiibo - IMG_4333

(f) M3-Amiibo - IMG_4356

Figure 1: Example images taken from each sequence of the `Amiibo` dataset. Top: example images from static sequences. Bottom: example images from dynamic sequences.

Toolbox using 20 shots of a 10×7 checkerboard pattern. The images used to calibrate the camera are included in the Supplementary Material for reference. Images are of size 4272×2848 (12 Megapixels) and the calibration matrix K is:

$$\mathsf{K} = \begin{pmatrix} 5461.35 & 0 & 2125.94 \\ 0 & 5468.33 & 1317.80 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

We also account for radial distortion in the images and report the two radial distortion coefficients of the lens k_1, k_2:

$$k_1 = 0.2230 \tag{2}$$

$$k_2 = 0.5958\,. \tag{3}$$

We did not consider a third distortion coefficient k_3 due to the fact that our camera exhibits very limited lens distortion. The distorted points are denoted as $(x_{\text{distorted}}, y_{\text{distorted}})$:

$$x_{\text{distorted}} = x(1 + k_1 * r^2 + k_2 * r^4) \tag{4}$$

$$y_{\text{distorted}} = y(1 + k_1 * r^2 + k_2 * r^4) \tag{5}$$

where $r = x^2 + y^2$. We use the radial distortion model above to undistort the images before extracting keypoints in the motion segmentation step.

Description of sequences The dataset comprises six sequences, with three static scene sequences and three dynamic scene sequences with either two or three motions. We provide a detailed description of each sequence:

(i) `S1-Amiibo` (16 images, static)

Depicts 3 static objects in slightly varying lighting conditions with a slightly out of focus background.

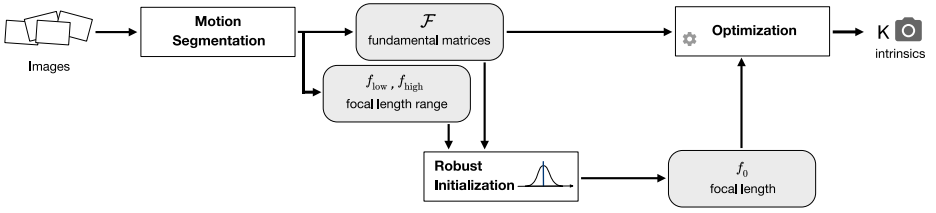


Figure 2: A graphical overview of the proposed multi-body self-calibration pipeline. The main steps of the algorithm are depicted (white boxes), as well as their inputs and outputs (grey boxes).

(ii) `S2-Amiibo` (12 images, static)

Depicts 3 static objects in addition to two planar surfaces in the background in slightly varying lighting conditions.

(iii) `S3-Amiibo` (17 images, static)

Depicts a large number of static small objects in varying lighting conditions and with a slightly out of focus background.

(iv) `M1-Amiibo` (10 images, dynamic)

Depicts a dynamic scene with a moving object (red bus) w.r.t. a static group of objects (green and yellow car) in varying lighting conditions. The background is mostly out of focus.

(v) `M2-Amiibo` (4 images, dynamic)

Depicts 3 moving objects in varying lighting conditions. The static element in the scene is given by 3 smaller objects near the center of the scene.

(vi) `M3-Amiibo` (4 images, dynamic)

Depicts 2 moving objects in consistent lighting conditions and with the background clearly in focus.

3 Experiments

This section provides additional experimental results to Sec. 4 of the main manuscript. Specifically, we include *i*) results from synthetic experiments referenced in Sec. 4.1 and Sec. 4.2 of the main paper, *ii*) the extended version of Tab. 1 of the main paper with the addition of execution times and the standard deviation of errors $\mathrm{err_f}$ and $\mathrm{err_{uv}}$. A scheme of the proposed method multi-body self-calibration pipeline is reported in Fig. 2.

3.1 Synthetic experiments

Following the structure of Sec. 4 of the main paper, we independently evaluate the *robust initialization* step, and then we assess the performance of the whole pipeline on synthetic datasets.

	vanilla		vanilla + KV		Our Init.	
$\#\mathcal{F}$	err_f	std_f	err_f	std_f	err_f	std_f
15	25.0	47.8	11.2	27.7	**3.9**	19.8
25	23.0	38.8	5.8	28.6	**1.05**	19.8
50	14.5	33.0	7.6	20.9	**2.80**	19.2
75	8.3	32.8	4.7	21.7	**1.06**	16.6

Table 1: Our robust initialization vs. alternative initialization strategies. Vanilla in green, Vanilla with Kernel Voting in red, and ours in blue. Number of fundamental matrices as input $\#\mathcal{F}$, relative error err_f and standard deviation std_f

3.1.1 Synthetic datasets

We generate a set of synthetic 3D scenes to establish 2D image matches. Each synthetic scene is made of several sphere-shaped objects defined by 100 3D points, which randomly translate and rotate. For every synthetic scene, we generate two projections matrices $\mathsf{P}_1 = [\mathsf{I}, \mathbf{t}_1]$ with $\mathbf{t}_1 = [0\ 0\ 1]^\top$, and $\mathsf{P}_2 = [\mathsf{R}_2, \mathbf{t}_2]$, where $\mathbf{t}_2$ is randomly sampled with a uniform distribution from 5 to 10, and R_2 corresponds to a random rotation along the yaw and pitch. We simulate image noise by adding zero-mean Gaussian noise with standard deviation $\sigma = 5$ to matches across both dimensions.

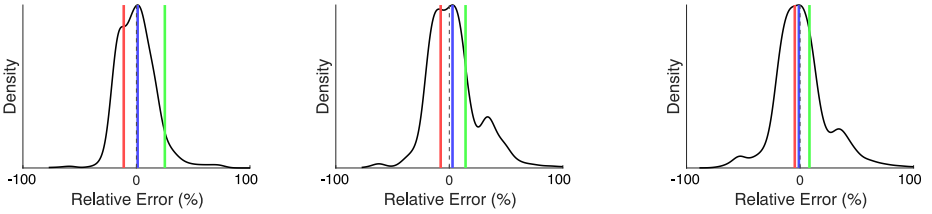
3.1.2 Robust initialization

We evaluate the accuracy err_f and the precision std_f, i.e., the standard deviation of err_f, of our robust initialization step on synthetic datasets against the same two baselines from the main manuscript: *i*) vanilla, the average of the focal lengths computed from each fundamental matrix using Sturm's method [5], ii) *vanilla with Kernel Voting*, where the vanilla method is equipped with Kernel Voting to select the most likely focal length. Specifically, our goal is to validate the claim from the main manuscript that our robust initialization step outperforms these baselines on synthetic, less-noisy datasets.

Tab. 1 compares our robust initialization against these baselines. We show that our solution achieves lower relative errors w.r.t. the real focal length and reduced std_f, which validates superior robustness. Fig. 3 shows the estimated kernel density functions $\hat{\phi}_f$ and the retrieved focals with a varying number of fundamental matrices provided as input. The x-axes are scaled to represent the relative error so that the origin corresponds to the ground-truth focal. Results show that the peak of the kernel density function produced by our method is always closer to the ground truth focal length (0% relative error) w.r.t. the two baselines. These results demonstrate that both the kernel voting and the exploration of maximal opening angles are beneficial and are responsible for improved precision and accuracy.

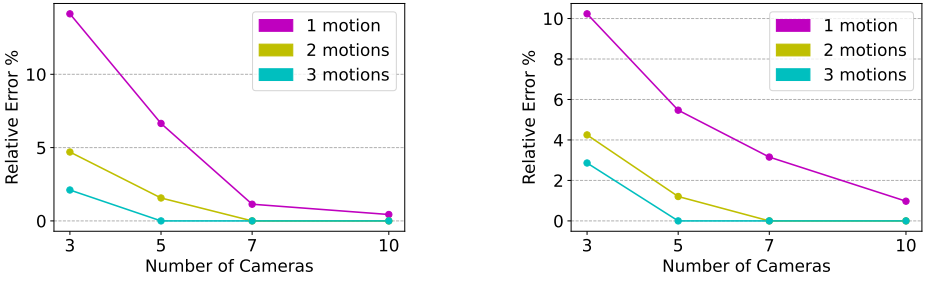
3.1.3 Complete pipeline

We evaluate our complete multi-body self-calibration pipeline on synthetic datasets characterized by either 1, 2 or 3 motions with a fixed percentage of outliers (50%), as described in Sec. 3.1.1. In this section, we validate the claims from the main manuscript that our method can successfully exploit the multiple motions in the scene to constrain the self-calibration better, and is robust towards noise and outliers on synthetic datasets as well.



(a) Synth. 15 fund (b) Synth. 50 fund (c) Synth. 75 fund

Figure 3: The kernel density function $\hat{\phi}_f$ plotted in black. x-axes are scaled to represent the relative error (%) $\mathrm{err_f}$ of the focal length. The green line is the result of vanilla, in red vanilla with Kernel Voting and in blue our robust initialization.



(a) $\mathrm{err_f}$, 50% outliers (b) $\mathrm{err_{uv}}$, 50% outliers

Figure 4: Relative error of our multi-body self-calibration configured with different number of motions varying the number of input images.

Fig. 4 shows $\mathrm{err_f}$ and $\mathrm{err_{uv}}$ as a function of the number of images in the synthetic sequences. Tests were repeated 100 times, and we reported the average errors from all the runs. Results confirm the findings from the main paper and show that our method is robust towards outliers and noise, as it successfully estimates the correct focal length and principal points when a sufficient number of views is available. Moreover, fewer images are needed to achieve accurate results as more moving bodies become available. Specifically, more bodies help in configurations where only a few images are provided as input. For instance, Fig.4 shows that using 3 motions in 5 images ($3 \times \binom{5}{2} = 30$ constraints) yields a relative error $\mathrm{err_f}$ which is lower than that achieved by using only 1 motion in 10 images, despite $\binom{10}{2} = 45$ constraints being available in the latter case.

3.2 Real experiments

In this section, we provide additional results from experiments on real datasets that were omitted from the main manuscript due to space constraints. First, in Sec. 3.2.1 we report and discuss the execution times of the experiments on real datasets. Then, in Sec. 3.2.2 we provide further results on critical motions and the exact results attained on the KITTI dataset [2].

			Single-body on KITTI sequences						Multi-body on KITTI sequences					
			Ours		BnB [4]		M&C [3]		Ours		BnB [4]		M&C [3]	
Dataset	m	$\lvert\mathcal{I}\rvert$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$	$\mathrm{err_f}$	$\mathrm{err_{uv}}$
Seq005	3	20	3.21	2.09	3.05	2.11	27.49	38.71	2.01	1.61	3.01	2.04	42.57	35.27
Seq048	2	20	56.01	39.26	42.89	43.16	65.41	47.13	55.47	43.98	64.61	56.71	78.63	93.51
Seq113	2	20	81.95	70.92	97.71	66.61	89.94	76.72	80.66	50.44	80.05	79.29	83.58	88.68

Table 2: Self-calibration comparison on the KITTI [2] dataset in Single-body and Multi-body. Number of rigid motions m in the dataset, total number of images processed $|\mathcal{I}|$, relative errors $\mathrm{err_f}$ and $\mathrm{err_{uv}}$. Tests were repeated 10 times, the average error is reported.

3.2.1 Extended results on timings

We report the extended version of Tab. 1 from the main paper with additional results on the standard deviation of $\mathrm{err_f}$ and $\mathrm{err_{uv}}$, and the running times of our method relative to the baselines of BnB [4] and M&C [3]. Execution times do not account for the Motion Segmentation step, which is common across all reported methods and takes a significant percentage of the total running time (at least 92% on average).

The reported execution times are averaged over 10 runs. Tests were performed on a 10-core Apple M1 Pro with 16 GB of RAM running MATLAB R2021b. BnB [4] and M&C [3] are also implemented in MATLAB. As reported in Sec. 4.2, our method is 60% slower on average than the robust BnB [4] method. However, the implementation of BnB relies on native (compiled) code for the expensive search paradigm that is significantly more efficient compared to an equivalent MATLAB implementation. In addition, performance from our method can be improved by reducing the number of iterations (at a slight cost in accuracy) or by using a more effective parallelization strategy for the multi-start. M&C [3] is non-robust and generally two orders of magnitude faster than our method and BnB.

3.2.2 Extended results on Critical Motions

As discussed in Sec. 4.2 of the main paper, we tested our self-calibration method on the KITTI dataset. Several sequences of the KITTI dataset represent a failure case for our method, as well as for BnB [4] and M&C [3]. It is well-known that critical configurations [6] limit the applicability of self-calibration. As explained in the main paper, the two-view self-calibration of a camera with a fixed focal length gives rise to critical configurations when the optical axes are parallel to each other. Specifically, this condition occurs when cameras move forward in a straight line. Forward motions are dominant in most sequences of the KITTI dataset [2] and our experiments show that we cannot calibrate the camera successfully. Tab. 2 reports experiments on three sequences of the KITTI dataset. For `Seq048` and `Seq113`, all methods produce significant errors $\mathrm{err_f}$ and $\mathrm{err_{uv}}$ and cannot calibrate the camera successfully. Instead, in `Seq005`, we can see the car turning several times and generally does not move in a straight line. In this scenario, our self-calibration method performs significantly better and can calibrate the camera.

| | | | Single-body on *static* datasets | | |

References

[1] https://github.com/andreadalcin/MultiBodySelfCalibration.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[3] P.R.S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:500–505, 1999.

[4] Danda Pani Paudel and Luc Van Gool. Sampling algebraic varieties for robust camera autocalibration. In *Proceedings of the European Conference on Computer Vision*, pages 265–281, 2018.

[5] P. Sturm. On focal length calibration from two views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 145–150, 2001.

[6] Peter Sturm. Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image and Vision Computing*, 20(5-6):415–426, 2002.