

Exciting Trajectories for Extrinsic Calibration of Mobile Robots with Cameras

Bogdan Khomutenko¹, Gaëtan Garcia¹, and Philippe Martinet¹

Abstract—The paper presents a method to precompute the optimal (so-called exciting) trajectories for the extrinsic calibration of a mobile robot equipped with the wheel odometry and one or more cameras. Considering the fact that the calibration is formulated as a non-linear least-square problem, the method is based on the analysis of the cost function properties in the neighborhood of the solution. By maximizing the determinant of the Hessian matrix, one makes the problem better defined and improves its robustness with respect to the measurement noise. Another convenience of the method is the possibility to reinforce additional constraints, like the visibility of a calibration object and the trajectory feasibility.

The source code of the application is publicly available as a part of `visgeom` project.

Index Terms—Camera calibration, Identification, Mobile robots

I. INTRODUCTION

Mobile robots equipped with cameras for localization and navigation purposes are becoming more and more popular. Since a robot is an object with a stable structure, its calibration seems to be a logical approach to reduce the number of unknowns during real-time computations. Intrinsic calibration of cameras may be considered as a solved problem. The Unified Camera Model [1] is used in most recent application involving wide-angle cameras. We use the enhanced version introduced in [2].

The problem of extrinsic calibration has been addressed in multiple papers; first for the manipulator equipped with eye-in-hand cameras (for example, [3]), and more recently for the mobile robots [4], [5], [6]. In [5] an impact of the trajectories on the calibration quality is mentioned, and a method to compute optimal trajectories for solving the linear part of the calibration, the wheel odometry intrinsic calibration, is proposed. Concerning the nonlinear part (which includes the camera extrinsic parameters) they give some intuitive guidelines. In particular, a straight trajectory does not give any information on the relative position of the camera with respect to the base, whereas a pure rotation makes the orientation of the sensor not fully observable.

We suggest a method to improve the calibration quality and its robustness with respect to the noise by computing the optimal trajectories. They provide the best possible definition of the calibration problem, which is formulated, in most cases, as a nonlinear optimization problem. The idea of generating exciting trajectories comes from the robotic arm community.

¹B. KHOMUTENKO, G. GARCIA, and P. MARTINET are with LS2N, Ecole Centrale de Nantes (ECN), Nantes, France `firstname.lastname@ls2n.fr`

It has been successfully applied to perform the dynamic identification [7], [8]. In [9] trajectory optimization for the extrinsic calibration of an eye-in-hand camera-manipulator system has been done. We present a framework which allows us to evaluate the quality of a given trajectory or of a set of trajectories for the extrinsic calibration. In this work the odometry is supposed to be calibrated and its output to be the integrated trajectory with the uncertainty.

The paper is structured as follows. In section II we present the calibration problem formulation. Note that it is general and does not rely on any trajectory properties. That is, for a given dataset it will do the best to estimate the unknowns. In section III we present the method of the trajectory evaluation and propose a cost function to optimize. Section IV gives few more technical detail about additional practical constraints on the trajectory as well as on the particular implementation. Section V demonstrates the numerical tests and results.

A. Notation

Across the paper, the following notation is used:

- \mathbf{p} — 2D point
- \mathbf{X} — 3D point
- ξ — 6 DoF transformation $[t_x, t_y, t_z, r_x, r_y, r_z]$, represented by translation and angle-axis rotation (\mathbf{t} and $\mathbf{r} = \mathbf{u}\theta$)
- $\xi(\mathbf{X})$ — transforming a 3D point \mathbf{X} from the frame defined by the transformation to its root frame.
- $\xi_a \circ \xi_b$ — composition of two transformations
- ζ_i — wheel odometry increment
- δ — infinitesimal motion
- \mathcal{C} — covariance matrix
- $\|\cdot\|_{\mathcal{C}}^2 : \mathbf{x} \mapsto \frac{1}{2} \mathbf{x}^T \mathcal{C}^{-1} \mathbf{x}$ — weighted 2-norm
- $[\cdot]_{\times}$ — cross-product skew-symmetric matrix

II. CALIBRATION PROBLEM FORMULATION

Camera intrinsics are supposed to be known. It is possible to integrate camera intrinsic calibration into this process, but we don't do it here for the sake of simplicity. The problem is formulated as a minimization of the following energy function:

$$E(\xi_c, \xi_b, \{\xi_i\}) = \sum_{i=1}^N \sum_{j=1}^M \|\mathbf{p}_{ij} - f^{(c,i)}(\mathbf{X}_j)\|_{\mathcal{C}_p}^2 + \sum_{i=1}^{N-1} \|\xi_{i+1}^{-1} \circ \xi_i \circ \zeta_i\|_{\mathcal{C}_{\zeta,i}}^2 \quad (1)$$

Hereafter N is the number of different robot's poses, M is the number of point features on the calibration board, p_{ij} is j^{th}

point feature, detected on i^{th} image. You can see the illustration in Fig. 1. The unknowns are ξ_c , ξ_b and $\{\xi_i\}$. The first term is responsible for the visual localization. $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a camera projection model. Before projecting, the point is transformed into the camera frame:

$${}^{c,i}\mathbf{X} = \xi_c^{-1} \circ \xi_i^{-1} \circ \xi_b({}^b\mathbf{X}) \quad (2)$$

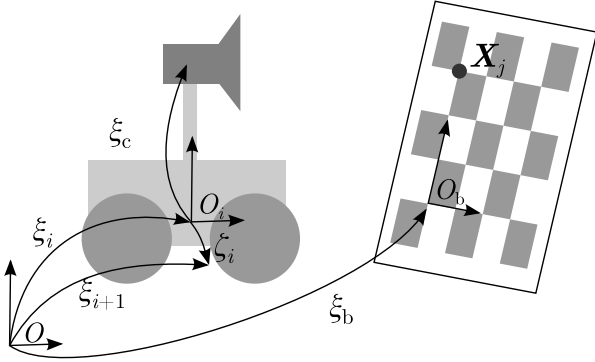


Fig. 1: A scheme of the calibration process. The unknowns are $\{\xi_i\}$, ξ_c , ξ_b . All of them take part in the computation of the projection of \mathbf{X} onto the image plane. Then the projection error is minimized. Another constraint is the loop $(\xi_i \circ \xi_i)^{-1} \circ \xi_{i+1}$ based on the odometry measurement ζ_i .

The second term penalizes the difference between each odometry increment ζ_i and the difference between the consecutive poses $\xi_i^{-1} \circ \xi_{i+1}$. Since the wheel odometry is noisy, we don't take the trajectory given by it directly, but compute the trajectory which respects as closely as possible each increment of the odometry. To get the values of the extrinsic parameters we solve the following problem:

$$\operatorname{argmin}_{\xi_c, \xi_b, \{\xi_i\}} E(\xi_c, \xi_b, \{\xi_i\}) \quad (3)$$

The only value that we actually need is ξ_c , but since ξ_b and $\{\xi_i\}$ are unknown we have to include it into the optimization problem.

It might be the case that we use multiple datasets to perform the calibration; in this case we will have one ξ_b and one $\{\xi_i\}$ for each of them, but it does not change the general concept.

III. OPTIMAL TRAJECTORY FOR THE EXTRINSIC CALIBRATION

To estimate the quality of the optimization problem defined in (3), we reformulate the problem. Instead of considering the coupled solution of the visual localization and odometry integration, which can be viewed as a SLAM problem, we say that the integrated odometry $\{\xi_{o,i}\}$ and the visual localization $\{\xi_{v,i}\}$ are given, as well as their covariance matrices ($\{C_{o,i}\}$ and $\{C_{v,i}\}$ respectively). It reduces the number of variables to ξ_c , but the uncertainty introduced by the calculus of the other

variables is taken into account. The reformulated problem looks as follows:

$$\begin{aligned} \xi_c^* &= \operatorname{argmin}_{\xi_c} E^* \\ E^* &= \sum_{i=1}^N \left\| \xi_c^{-1} \circ \xi_{o,i}^{-1} \circ \xi_c \circ \xi_{v,i} \right\|_{C_i}^2 \\ &+ \left\| \hat{\xi}_c^{-1} \circ \xi_c \right\|_{C_{\hat{\xi}_c}}^2 \end{aligned} \quad (4)$$

where $\hat{\xi}_c$ is the prior estimation of the extrinsics; $C_i = C_{o,i} + C_{v,i}$ is the covariance defining the measurement noise for the position; $\xi_{o,i}$ is measured with the wheel odometry; and $\xi_{v,i}$ is the localization of the calibrated camera using a calibration board. The first term means that, as shown on Fig. 2, we want:

$$\xi_{o,i} \circ \xi_c = \xi_c \circ \xi_{v,i} \quad (5)$$

We want to analyze the properties of the energy function E^* from (4) in the neighborhood of the solution. To improve the problem stability properties we can minimize the following criterion:

$$F = - \sum_{i=1}^6 \log \sigma_i \left(\frac{\partial^2}{\partial \xi_c^2} E \right) \quad (6)$$

where $\sigma_i(\cdot)$ is the i^{th} singular value of a matrix. Basically, it is equivalent to maximization of the Hessian matrix determinant $\det(H) = \prod_{i=1}^6 \sigma_i(H)$. In [9] they suggest the matrix condition number as a criterion instead of the determinant, in [7] one over the minimum singular values is proposed. But in our case one degree of freedom is not observable, hence the corresponding singular value is defined only via the prior transformation estimation and does not depend on the trajectory. In fact, the condition number will be just proportional to the maximum singular value. Yet, there might be a better criterion, than (6), like a different function of the singular values; it is still an open question.

The Hessian matrix itself $\frac{\partial^2}{\partial \xi_c^2} E$ can be used to analyze the observability of different degrees of freedom. Its singular decomposition and singular vectors corresponding to small or zero singular values tell us which degrees of freedom of the extrinsic transformation cannot be observed.

In our case (3 DoF mobile robot) the only rotation is about z -axis, hence z -component of the translation of ξ_c is not observable (as it is mentioned in [6]), and its value is defined by the prior transformation estimation. But x , y , and $u\theta$ are observable, and the larger the singular values, the better the final numeric estimation.

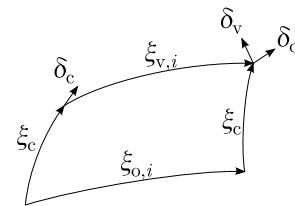


Fig. 2: δ_c is an infinitesimal increment of ξ_c , notice that δ_c appears twice since ξ_c is present twice in the transformation chain; δ_v is the increment, transformed into the terminal frame; $\delta = \delta_v - \delta_c$ is the residual which appears when we modify the solution ξ_c by δ_c .

If covariances are defined for the infinitesimal motions of the terminal frame, then we need to transform the infinitesimal increments δ_c of ξ_c into the terminal frame (Fig. 2):

$$\delta_v = {}^{v,i}\mathbf{T}_{v,1}\delta_c \quad (7)$$

$${}^{v,i}\mathbf{T}_{v,1} = \begin{pmatrix} R_{v,i}^{-1} & -R_{v,i}^{-1}[\mathbf{t}_{v,i}]_{\times} \\ 0 & R_{v,i}^{-1} \end{pmatrix} \quad (8)$$

where ${}^i\mathbf{T}_j$ is a twist transformation matrix, $R_{v,i}$ and $\mathbf{t}_{v,i}$ is another representation of $\xi_{v,i}$. Another way of writing the residual in energy expression (4) is:

$$\delta = \delta_v - \delta_c \approx \xi_c^{-1} \circ \xi_{v,i}^{-1} \circ \xi_c \circ \xi_{v,i} \quad (9)$$

A. Covariance for the Visual Localization

Here we assume that the visual localization is done using a calibration board. Any other visual localization can be used to compute the trajectory of the camera, but the similar precision analysis must be done in this case. The covariance of the visual localization, assuming the camera intrinsic parameters exactly known, comes from the uncertainty in the corner detections of the calibration board. The localization is computed as non-linear least squares. The cost function is defined as:

$$E_v = \sum_{i=1}^M \|\mathbf{p}_i - f(\mathbf{X}_i, \xi)\|_{\mathcal{C}_p}^2 \quad (10)$$

Here \mathcal{C}_p is a 2×2 corner detection covariance matrix. We assume that the coordinates of a detected feature point are decorrelated and have equal uncertainties:

$$\mathcal{C}_p = \begin{pmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_p^2 \end{pmatrix} = \sigma_p^2 I \quad (11)$$

For practical purposes we can assume that $\sigma_p = 1$, unless we have a better prior estimation of corner detector precision.

The whole least-squares problem for all points together is defined as follows:

$$J^T \mathcal{C}_{\Delta p}^{-1} \Delta p = 0 \quad (12)$$

where J is a concatenation of Jacobian matrices for individual points:

$$J = \begin{pmatrix} \frac{\partial f(\xi(\mathbf{X}_1))}{\partial \xi} \\ \vdots \\ \frac{\partial f(\xi(\mathbf{X}_M))}{\partial \xi} \end{pmatrix} = \begin{pmatrix} J_1 \\ \vdots \\ J_M \end{pmatrix} \quad (13)$$

Δp is the concatenation of reprojection errors:

$$\Delta p = \begin{pmatrix} \mathbf{p}_1 - f(\mathbf{X}_1, \xi) \\ \vdots \\ \mathbf{p}_M - f(\mathbf{X}_M, \xi) \end{pmatrix} \quad (14)$$

$\mathcal{C}_{\Delta p}^{-1}$ is the covariance matrix of Δp :

$$\mathcal{C}_{\Delta p}^{-1} = \begin{pmatrix} \mathcal{C}_p^{-1} & 0 & \dots & 0 \\ 0 & \mathcal{C}_p^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathcal{C}_p^{-1} \end{pmatrix} \quad (15)$$

Let us consider that the solution for (12) is found by using the iterative Newton-Gauss optimization. The solution looks as follows (for the last iteration, [10]):

$$\Delta \xi = \underbrace{(J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} J^T \mathcal{C}_{\Delta p}^{-1} \Delta p}_{J_c^+} \quad (16)$$

Here, J_c^+ is the weighted generalized inverse of J . We know that if a linear mapping is applied to a random Gaussian vector:

$$\mathbf{y} = A\mathbf{x} \quad (17)$$

Then the covariance of the result is computed as:

$$\mathcal{C}_y = A\mathcal{C}_x A^T \quad (18)$$

In our case:

$$\begin{aligned} \mathcal{C}_v &= J_c^+ \mathcal{C}_{\Delta p}^{-1} J_c^{+T} \\ &= (J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} J^T \mathcal{C}_{\Delta p}^{-1} \mathcal{C}_{\Delta p} \mathcal{C}_{\Delta p}^{-1} J (J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} \\ &= (J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} J^T \mathcal{C}_{\Delta p}^{-1} J (J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} \\ &= (J^T \mathcal{C}_{\Delta p}^{-1} J)^{-1} \end{aligned} \quad (19)$$

This general form, which is an interesting theoretical result by itself, can be simplified even further, if we recall that $\mathcal{C}_{\Delta p}^{-1} = \sigma_p^{-2} I$:

$$\mathcal{C}_v = \sigma_p^2 (J^T J)^{-1} \quad (20)$$

B. Odometry Covariance Matrix

We assume the unicycle kinematic model, locally circular motion. For a different kind of kinematic scheme, strictly speaking, it might be necessary to do a similar analysis. But in our belief, it will not make much difference. For the sake of clarity, to avoid introducing additional notation, we denote the 3 DoF motion increment by ζ , since the other DoFs equal zero.

$$\zeta_i = \begin{pmatrix} x \\ y \\ \varphi \end{pmatrix} = \begin{pmatrix} l \cos \frac{\varphi}{2} \\ l \sin \frac{\varphi}{2} \\ \varphi \end{pmatrix} \quad (21)$$

where l is the elementary distance and φ is the elementary rotation (Fig. 3). Together they form a control vector \mathbf{u} .

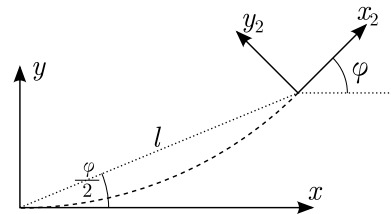


Fig. 3: The circular motion model. φ is the turning angle; l is the distance. There are two frames: xy — the starting frame, x_2y_2 — the terminal local frame. The motion here is exaggerated to illustrate the geometry, but in reality the model is applied for small translations and rotations.

We are interested in $\mathcal{C}_{\zeta,i}$, the covariance which characterizes the error distribution of the odometry increment measurement in its local frame. It can be computed as follows:

$$\mathcal{C}_{\zeta,i} = \frac{\partial \zeta_{i,\text{local}}}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial \mathbf{u}} \mathcal{C}_{\mathbf{u},i} \left(\frac{\partial \zeta_{i,\text{local}}}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial \mathbf{u}} \right)^T \quad (22)$$

Here $\mathcal{C}_{\mathbf{u},i}$ is the covariance of the control vector; $\zeta_{i,\text{local}}$ is the elementary motion expressed in the terminal frame.

The transformation between the root frame and the local frame is:

$$\frac{\partial \zeta_{i,\text{local}}}{\partial \zeta_i} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (23)$$

The Jacobian matrix of the motion:

$$\frac{\partial \zeta_i}{\partial \mathbf{u}} = \begin{pmatrix} \cos \frac{\varphi}{2} & -\frac{l}{2} \sin \frac{\varphi}{2} \\ \sin \frac{\varphi}{2} & \frac{l}{2} \cos \frac{\varphi}{2} \\ 0 & 1 \end{pmatrix} \quad (24)$$

Multiplying the two matrices yields:

$$\frac{\partial \zeta_{i,\text{local}}}{\partial \zeta_i} \frac{\partial \zeta_i}{\partial \mathbf{u}} = \begin{pmatrix} \cos \frac{\varphi}{2} & \frac{l}{2} \sin \frac{\varphi}{2} \\ -\sin \frac{\varphi}{2} & \frac{l}{2} \cos \frac{\varphi}{2} \\ 0 & 1 \end{pmatrix} \quad (25)$$

We need to integrate the covariance matrix along the trajectory to get its actual value at a given point:

$${}^{o,i}\mathcal{C}_{o,i} = \mathcal{C}_{\zeta,i} + {}^i\mathbf{T}_{i-1} {}^{o,i-1}\mathcal{C}_{o,i-1} {}^i\mathbf{T}_{i-1}^T \quad (26)$$

\mathbf{T} is a twist transformation matrix, like in (8). The given expression gives a matrix 3×3 ; three more rows and columns are to be added to make it 6×6 . We also add a small constant covariance matrix $\tilde{\mathcal{C}}$ to it, to introduce some uncertainty in the other 3 directions, then the frame must be changed from i -th odometry frame to i -th camera frame:

$$\mathcal{C}_{o,i} = {}^v,i\mathbf{T}_{o,i} ({}^{o,i}\mathcal{C}_{o,i} + \tilde{\mathcal{C}}) {}^v,i\mathbf{T}_{o,i}^T \quad (27)$$

IV. TECHNICAL DETAILS

The camera model used here is the Enhanced Unified Camera Model, presented in [2]. We used the *Ceres solver* [11] as a non-linear solver for both trajectory optimization and extrinsic calibration problems. The source code of the project can be found at <https://github.com/BKhomutenko/visgeom>.

As it is mentioned in [5], in any experimental setup there are some practical constraints which are not necessarily represented by the cost function (6). It is possible to augment it by adding terms to find the best of practical and feasible trajectories.

A. Visibility Constraint

To make sure that the images contain the calibration board along the trajectory, we reinforce the visibility constraint by adding a term to (6):

$$F = - \sum_{i=1}^6 \log \sigma_i \left(\frac{\partial^2}{\partial \xi_c^2} E \right) + \sum_{j=1}^M \sum_{k=1}^N \rho_r (f(\mathbf{X}_j, \xi_k, \xi_c)) \quad (28)$$

here $\rho_r : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the penalty functions. it works as follows:

$$\rho_r(\mathbf{p}) = \begin{cases} 0 & \text{if } \|\mathbf{p} - \mathbf{p}_0\| < r \\ \lambda(\|\mathbf{p} - \mathbf{p}_0\| - r)^2 & \text{otherwise} \end{cases} \quad (29)$$

where \mathbf{p}_0 is the center of projection; r is a certain radius. We can divide image into three zones, as it is shown on Fig. 4. All the feature points of the calibration board in the first zone are not penalized (and its radius is r), but once a point approaches zone 2, the additional cost starts growing quadratically, which means, considering that the data term is a logarithm of singular values, that all the points will rather stay in zone 1 after the optimization.

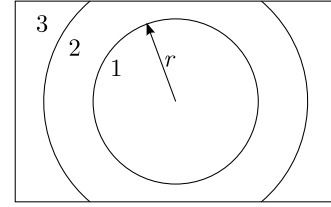


Fig. 4: Every image is divided into three concentric zones: 1) the additional cost for the points there is 0; 2) the additional cost for points grows quadratically with respect to the distance from the projection center; 3) there are no points projected into this zone because of the fact that we are using fisheye cameras.

This method is adapted for the fisheye optics. For pinhole cameras it might be interesting to use a different regularization term.

B. Constraints on the Trajectory Curvature

For certain types of mobile robots (such as ones with steerable wheels) there are limits on the trajectory curvature. This constraint can be taken into account by adding the following term to the cost function:

$$E_{\text{curvature}} = \lambda_{\kappa} \sum_{i=1}^{N-1} \max \left(\frac{\varphi_i}{l_i} - \kappa_{\max}, 0 \right)^2 \quad (30)$$

Where φ is the elementary rotation, l is the elementary translation length.

C. Trajectory Parametrization

It appears that a trajectory which is a single arc of a constant curvature is not enough to do the extrinsic identification. It has a simple explanation: for such a trajectory the camera's trajectory will also be an arc of a different radius. If we rotate the camera's initial position and the calibration board about the center of the circular trajectory, then the camera's

motion will be the same with respect to the calibration board. It means that such a trajectory will define a circle with the center at the trajectory's center, on which lies the camera. If we take two distinct arcs with different curvature as a trajectory, it will define two circles, and the camera has to be at one of the intersections, since two circles generate two intersections. If the camera's prior position is accurate enough, the optimization will converge to its real position, that is why the trajectory is represented by two arcs. It is parametrized by their initial point along x , initial orientation, linear and angular speed of the robot. It makes 8 optimization parameters in total. "The optimal trajectory" refers to the solution of the optimization problem with such a parametrization.

D. Optimization Initialization

A good initialization is important to have the problem converged to the global optimum. In order to do that the initialization is carried out as follows:

- 1) The prior on the camera extrinsics ξ_c must be provided by the user. Generally, its position can be measured with a reasonable precision, the orientation prior can be approximative.
- 2) Transformations $\{\xi_i\}$ are initialized using the odometry measurements, which is straight-forward.
- 3) ξ_b is initialized in a few steps:
 - a) Using first image of a sequence, compute ${}^c\xi_b$ via the virtual visual servoing.
 - b) Compute $\hat{\xi}_b = \xi_1 \circ \xi_c \circ {}^c\xi_b$
 - c) Using all the images of the sequence, minimize the reprojection residual over ξ_b , using $\hat{\xi}_b$ as the initialization. Basically we minimize (1) fixing all the variables, except for ξ_b .

The camera intrinsics are supposed to be known. Another option is to get their accurate estimation and include one more block into the optimization problem along with a set of images to perform the intrinsic calibration simultaneously with the extrinsic one.

V. RESULTS

The concept has been tested using synthetic data, which allows us to compare the results with the true value. That is, for a generated trajectory, a set of images of the calibration board has been generated, using the camera model (30 images per trajectory piece). These images have been used to perform the calibration. Even though the images are synthetic, the image processing part still introduces a certain error in the calibration due to the imperfection of the pattern detection. The extrinsic calibration has been done twice: once using the optimal trajectory, and one more time using a suboptimal one. The suboptimal trajectory has been obtained by stopping the optimization process before the convergence, but after the visibility and curvature constraints have been satisfied. You can see both trajectories on the Fig. 5.

To check the extrinsic calibration robustness, some noise has been added to the odometry measurements. The absolute value of the noise and the resulting calibration error

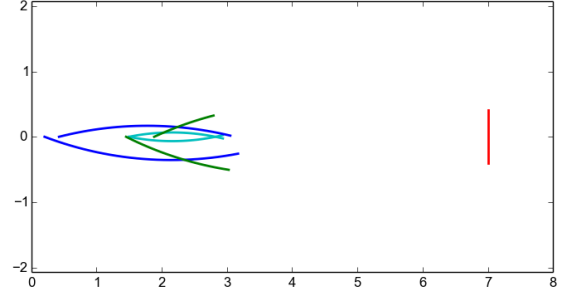


Fig. 5: Generated trajectories: Cyan – the initial trajectory, $E = 688$, the curvature constraint is not satisfied; green – a suboptimal trajectory, $E = -29.6$; Blue – the optimal trajectory, $E = -37.4$. E is the cost function.

TABLE I: Comparison between extrinsic calibration quality for two trajectories. The numbers in the last four columns represent the error in the extrinsic transformation estimation.

Odometry noise, absolute value		Optimal trajectory		Suboptimal trajectory	
t , m	$u\theta$, rad	e_t , m	e_θ , rad	e_t , m	e_θ , rad
0	0	0.00935	0.000665	0.0166	0.00181
0.002	0.01	0.0101	0.00302	0.0431	0.0143
0.004	0.02	0.0164	0.00727	0.199	0.0507
0.006	0.03	0.0628	0.0149	0.436	0.105
0.008	0.04	0.13	0.0238	0.739	0.17

are summarized in Table I. The results are also visualized on Fig. 6. The true values of the extrinsic parameters are $\xi_c = [3.5, 0.35, 0, -1.2092, 1.2092, -1.2092]$. The errors are computed in the following way:

$$\begin{aligned} \delta &= \xi_c^{-1} \circ \hat{\xi}_c \\ e_t &= \|\mathbf{t}_\delta\| \\ e_\theta &= \theta_\delta \end{aligned} \quad (31)$$

where $\hat{\xi}_c$ is the estimated extrinsic parameters; \mathbf{t}_δ is their translational part; θ is the norm of the rotation vector of δ

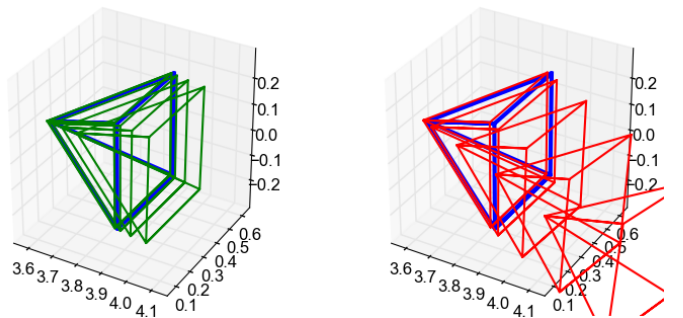


Fig. 6: Visualization of the calibration results as a function of the odometry noise. Blue – ground truth, green – optimal trajectory, blue – suboptimal trajectory. The camera wire model is a pyramid $0.4 \times 0.4 \times 0.4$ m. The stronger the noise, the farther away the estimated extrinsics from the real values. You can see that for the optimal trajectory (on the left) the sensitivity to the noise is significantly smaller than for the suboptimal one.

You can see that the optimal trajectory is significantly more resistant to the noise in the odometry measurements. The noise in the test has zero mean, and in our strong belief odometry

TABLE II: Comparison between extrinsic calibration quality for monocular and stereo systems. The numbers in the last four columns represent the error in the extrinsic transformation estimation. The transformation is the same in both cases.

Odometry noise, absolute value		Monocular		Stereo	
t , m	$u\theta$, rad	e_t , m	e_θ , rad	e_t , m	e_θ , rad
0	0	0.00935	0.000665	0.0172	0.00123
0.002	0.01	0.0101	0.00302	0.0102	0.00331
0.004	0.02	0.0164	0.00727	0.0173	0.00506
0.006	0.03	0.0628	0.0149	0.0565	0.00804
0.008	0.04	0.13	0.0238	0.116	0.0131

bias would have a very negative effect on this calibration. It must be eliminated during the preceding odometry calibration.

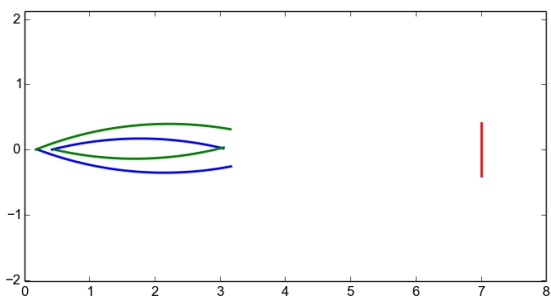


Fig. 7: Generated trajectories for the stereo extrinsic calibration. Blue – the optimal trajectory for the left camera, green – Blue – the optimal trajectory for the right camera, red – the calibration board position. Considering the symmetry of the stereo system, it is logical that the two trajectory sets are symmetrical.

We can perform simultaneous calibration of extrinsic parameters for a stereo system. For that we define the transformation between the board and the cameras as follows:

$$\begin{aligned} {}^c_1\xi_b &= (\xi_{0,i} \circ \xi_c)^{-1} \circ \xi_b \\ {}^c_2\xi_b &= (\xi_{0,i} \circ \xi_c \circ {}^c_1\xi_{c_2})^{-1} \circ \xi_b \end{aligned} \quad (32)$$

The extrinsic parameters for the first camera are the same. For the second $\xi_{c_2} = [3.5, -0.35, 0, -1.2092, 1.2092, -1.2092]$.

Since extrinsic transformations for the two cameras are different, different trajectories have to be used for the optimal calibration. The trajectories are represented on Fig. 7. In the optimization problem we have the following cost function blocks:

- 1) The first camera optimal trajectory — 60 images.
- 2) The second camera optimal trajectory — 60 images.
- 3) The stereo calibration — two times 27 images.

You can see the results in Table II. For stronger noise the stereo version gives a slightly better orientation estimation, but generally the order of precision is the same as for the monocular calibration. Note that we generated the calibration trajectories regardless the fact that the two cameras' extrinsics are coupled via the stereo calibration.

VI. CONCLUSION

Planned data acquisition can improve the calibration quality. The developed methodology lets us evaluate a trajectory's

quality for the extrinsic calibration, taking into account the odometry noise properties and image-based localization uncertainty. Global optimization problem allows us to include additional constraints, such as space limitations, size of the calibration board in the image and so on.

The synthetic data allowed us to test the concept in a fully controlled environment; the next step is to perform the extrinsic calibration of real experimental platforms, and validate its precision via visual localization algorithms.

The criterion on the trajectory quality for the extrinsic calibration can be applied for any kind of visual localization, whether they use the calibration board or not. Here we suggest a methodology to estimate the localization precision using the calibration board, but any visual odometry system can replace it (For example, [6]). The only requirement is to be able to estimate the localization covariance matrix. For the natural features doing it precisely is a challenging problem. A possible research track might be to estimate the average visual localization precision statistically. Another option is to do it in two steps. First using any kind of trajectory perform visual SLAM, then use a similar calculations as in Section III.A to get the localization covariance estimation, based on the environmental structure. Analyzing the features visibility is also a complicated problem.

The method can also be used to analyze the observability properties of extrinsic parameters for a given trajectory. Another interesting extension might be to compute the optimal trajectories for the simultaneous wheel odometry, IMU, and exteroceptive sensor calibration.

REFERENCES

- [1] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *IEEE International Conference on Robotics and Automation*, April 2007, pp. 3945–3950.
- [2] B. Khomutenko, G. Garcia, and P. Martinet, "An enhanced unified camera model," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 137–144, Jan 2016.
- [3] K. Daniilidis, "Hand-eye calibration using dual quaternions," *International Journal of Robotics Research*, vol. 18, pp. 286–298, 1998.
- [4] G. Antonelli, F. Caccavale, F. Grossi, and A. Marino, "Simultaneous calibration of odometry and camera for a differential drive mobile robot," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 5417–5422.
- [5] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo, "Simultaneous calibration of odometry and sensor parameters for mobile robots," *IEEE Transaction on Robotics*, vol. 29, pp. 475–492, 04/2013 2013.
- [6] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1793–1800.
- [7] B. Armstrong, "On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics," *The International Journal of Robotics Research*, vol. 8, no. 6, pp. 28–48, 1989.
- [8] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," in *[30th IEEE Conference on Decision and Control*, vol. 1, Dec 1991, pp. 494–499.
- [9] P. Renaud, N. Andreff, G. Gogu, and M. Dhome, "Optimal pose selection for vision-based kinematic calibration of parallel mechanisms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, Oct 2003, pp. 2223–2228.
- [10] M. I. A. Lourakis and A. A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," *FORTH-ICS*, Tech. Rep. 340, 2004.
- [11] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>.