2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Efficient Pose Prediction with Rational Regression applied to vSLAM

George Terzakis[1] and Manolis Lourakis[2]

Abstract— **Compared to polynomial splines, rational functions are known to be more efficient and well-behaved data fitting models. However, due to the potential presence of zeros in their denominator, rational functions tend to yield notoriously hard optimization problems. In this work, we present a novel least squares method for 6D pose prediction that employs rational regression. Our method can accommodate fixed data points and is able to circumvent the occurrence of zeros for rational quadratic interpolants. We demonstrate the suitability of rational quadratics for pose prediction by applying our approach to real data from the feature tracking stage of a real-time visual SLAM system and showing that it yields far more stable predictions when compared to state-of-the-art rational and polynomial spline methods.**

I. Introduction

Predicting camera motion is a frequent task in real-time structure from motion (SfM) applications, e.g. [1], [2], [3], [4], [5], [6]. In the majority of cases, such applications are designed to operate in an agnostic manner with respect to the nature of camera motion in order to be independent of the platform the camera is attached to. Leveraging previously estimated poses to obtain a reasonably stable estimate of present camera pose allows the tracking system to anticipate where known features from the the previous frame will have moved in the present one. Therefore, pose prediction is important for successful preliminary feature tracking.

In purely visual simultaneous localization and mapping (SLAM), simple techniques for pose prediction relying on the relative pose between the two most recent captured frames have been employed in the tracking front-end, e.g., PTAM [7] or ORB-SLAM [8]. In the case of discrete time visual-inertial SLAM (viSLAM), more complex pose prediction models that implicitly use quadratic position polynomials and linear orientation increments between inertial measurement unit (IMU) samples have been employed, e.g. [9], [10], [11]. More elaborate polynomial spline schemes were proposed in the context of continuous-time visual SLAM (vSLAM) with optional inclusion of IMU measurements [12], [5], [13].

The rest of this section presents an overview of existing relevant works from the literature. The novel aspects of the proposed method are outlined in Section II whereas its details are presented in Section III. Experimental evaluation results are reported in Section IV and a conclusion is in Section V.

[1]George Terzakis is with the Institute of Communication & Computer Systems (ICCS), 9 Ir. Polytechneiou, Zografou, 15773, Athens, Greece. george.terzakis@iccs.gr

[2]Manolis Lourakis is with the Institute of Computer Science, Foundation for Research and Technology – Hellas, P.O. Box 1385, 711 10, Heraklion, Crete, Greece. lourakis@ics.forth.gr

A. Related work

Polynomial splines have been employed not only for pose prediction during tracking, but also to fully parametrize pose in the back-end non-linear optimization in continuous-time vSLAM. In [6] and [14], a continuous-time parametrization of pose with polynomial basis functions is proposed along with a novel visual-inertial optimization framework. Instead of targeting pose variables, it focuses on the coefficients of the corresponding basis functions. This approach was adopted and extended with time derivative regularization terms in [5] and [12]. In the latter study, the use of IMU is optional, wherein the "missing" constraints on velocity acceleration and angular velocity can be substituted by regularization terms. The resulting polynomial splines yield pose predictions that primarily enhance the performance of tracking but they also constitute an alternative parametrization for camera pose that is refined in the context of iterative optimization of the reprojection error.

Non-parametric regression with Gaussian Processes (GP) is the other side of the pose regression coin. In [15], [16], [13], the entire pose optimization framework is approached with GP pose priors. An important downside in this methodology is the scaling of the GP covariance matrix, particularly when trying to employ the prior in a non-linear optimization framework that includes visual landmark observations. Further, Persson et al. [12] argue that another shortcoming of the use of GP pose priors compared to splines is that they induce short optimization windows in orientation due to approximations. Note, however, that for the purpose of predictions from previous poses during tracking, a GP prior is a plausible regression technique and we therefore include it in our comparisons in Section IV.

B. Rational regression

Polynomial approximations suffer from the so-called Runge's phenomenon [17], [18], which refers to the tendency of the interpolant to oscillate away from data points as the degree of the polynomial increases, like demonstrated in Figure 1. Thus, predictions with polynomial splines can become unreliable due to this issue. In contrast, rational models that represent a given function as the quotient of two polynomials, are known to remain more faithful to the trend of the data and therefore possess good predictive (i.e., extrapolation) properties [19], [20].

Despite their attractive properties, rational functions present significant challenges in regression such as the accidental occurrence of unwanted zeros (aka poles) in the denominator as a by-product of a potentially naive data-fitting algorithm, or the difficulty of controlling derivatives

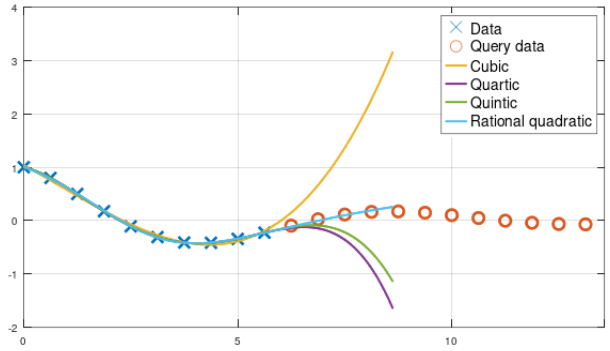

Fig. 1. Polynomials of 3rd, 4th, 5th degree and rational quadratic fit to data points drawn from $y = \exp(-0.2t)\cos(-0.7t)$ in the interval $[0,\ 6]$. All polynomial curves diverge at the right edge of the interval manifesting Runge's phenomenon, whereas the rational quadratic (cyan) remains more faithful to the trend of the curve.

in ways that are very forthcoming in the case of polynomial splines. Floater and Hormann [21] proposed a family of barycentric rational interpolants that lack real poles and have arbitrarily high approximation orders on any real interval, regardless of the distribution of the points. The techniques employed to avoid the unwanted poles are typically numerical and recursive in nature since they re-estimate the model as they sequentially parse the data points [22], [20], [23]. Such a technique is the AAA algorithm [20], which builds the interpolating rational function incrementally. In particular, AAA repeatedly recovers sets of weights via least squares in a series of steps as it progressively builds a set of support points from the data. The poles are chosen in a fashion that the singularities that emerge are benign. The way this works is to choose the next support point by identifying the corresponding weight associated with the highest residual in the previous step.

II. Contributions

This work puts forward a novel method for pose regression and interpolation with a rational quadratic, i.e., a ratio of polynomials of degree at most two. The method was devised in order to efficiently predict the orientation and location of a coordinate frame by fitting or interpolating where necessary, a sequence of poses previously estimated (e.g., by means of SLAM/vSLAM). To the best of our knowledge, this is the first time that a rational approximant/interpolant is used for pose prediction particularly for SLAM/vSLAM applications. The novelties can be summarized in the following:

1) We devise a linear least squares (LS) algorithm that performs regression with optional fixed data points, using a 5-degree of freedom (DoF) model that is guaranteed pole-free.
2) The algorithm has characteristics suitable for pose regression in real-time applications. Specifically, being a linear LS solver, it exhibits very low computational overhead. Furthermore, with 5 DoF, it can fit more than 5 past poses[1], a number sufficiently large to maximize the prediction's efficiency for standard frame rates.
3) It is straightforward to implement. The solution of the regression problem involves operations with up to 3×3 matrices, thereby allowing for the use of fast analytical methods that can be deployed even on low capacity hardware [24]. Our C++ implementation is publicly available.

III. Method

We define our pose prediction problem with an outlook to the tracking front-end of a vSLAM pipeline such as [7], [25], [8], [10]. In such a setup, a number of past pose estimates is available, typically with no motion prior. Further, we assume no additional process input such as IMU readings[2]. We will construct a rational quadratic that fits the data and interpolates up to 4 data points (6D pose vectors).

A. Pose convention and parametrization

Let $\mathrm{SE}(3)$ be the special Euclidean group consisting of all proper rigid transformations in the 3D Cartesian space. Denote n past pose estimates by $\boldsymbol{T}_1,\ldots,\boldsymbol{T}_n \in \mathrm{SE}(3)$ at times $t_1,\ldots,t_n$ and the goal is to predict the pose of the most recently captured image frame. We choose to fit pose in its decoupled orientation – position representation, i.e., as a body-to-world point transformation as suggested by Ovren et al. [5]. In other words, the pose $\boldsymbol{T}(t)$ as a 4x4 homogenous matrix is

$$\boldsymbol{T}(t) = \begin{bmatrix} \boldsymbol{R}(t) & \boldsymbol{p}(t) \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{R}(t)$ has the direction vectors of the local frame as its columns and $\boldsymbol{p}(t)$ is the position of the camera in the world. Ovren et al. [5] argue that this convention decouples the location from the orientation and therefore favors more efficient pose fitting.

Clearly, position vectors $\boldsymbol{p}_1,\ldots,\boldsymbol{p}_n$ can be used directly in the LS formulation to fit the position function. However, in the case of orientation, a minimal DoF parametrization is necessary. We thus choose the modified Rodrigues parameters (MRPs) [26], [27], [28] to represent orientation. Denote next the orientation MRPs at the data points as $\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_n \in \mathbb{R}^3$ and the corresponding function in time, $\boldsymbol{\psi}(t)$. Like any other minimal parametrization of orientation, MRPs present a singularity on the quaternion sphere, yet it is a benign one. This is because not only the corresponding rotation matrix is representable at the origin, but in fact, all rotations can be represented by an MRP vector $\boldsymbol{\psi}$, such that $\|\boldsymbol{\psi}\| < 1$[3]. For completeness, we provide below the expressions that relate MRPs with the corresponding axis-angle vector $\boldsymbol{\theta}$ and quaternion $q = (s,\ \boldsymbol{v})$ as given in [29]:

$$\boldsymbol{\psi} = \tan\left(\frac{\|\boldsymbol{\theta}\|}{4}\right)\frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|} = \frac{\boldsymbol{v}}{1+s}. \tag{2}$$

[1]The typical choice in our experiments was 8 with the 2 most recent ones being interpolated/fixed.

[2]The method can be extended to also account for IMU samples.

[3]MRPs inside the unit sphere in $\mathbb{R}^3$ map to exactly one hemisphere of the quaternion sphere.

B. Rational quadratic regression

Consider $n \geq 5$ samples $y_1 \ldots, y_n \in \mathbb{R}$ at the corresponding time instances $t_1, \ldots, t_n$, such that $t_1 < \cdots < t_n$. We wish to fit this data to the rational function

$$f(t) = \frac{a_0 + a_1 t + a_2 t^2}{b_0 + b_1 t + b_2 t^2} = \frac{\boldsymbol{a}^T \boldsymbol{\tau}(t)}{\boldsymbol{b}^T \boldsymbol{\tau}(t)}, \tag{3}$$

where $\boldsymbol{a} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^T$, $\boldsymbol{b} = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix}^T$ and $\boldsymbol{\tau}(t) = \begin{bmatrix} 1 & t & t^2 \end{bmatrix}^T$, subject to the constraint that the denominator has no zeros, i.e. its discriminant is negative:

$$b_1^2 - 4 b_0 b_2 = \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b} < 0, \;\; \boldsymbol{K} = \begin{bmatrix} 0 & 0 & -2 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}. \tag{4}$$

1) Simple approximation: We begin our analysis by considering the problem of approximating the data points with the rational function f in (3). This is a quadratic program with a data likelihood LS objective and a quadratic constraint corresponding to the negative discriminant of (4),

$$\underset{\boldsymbol{a},\, \boldsymbol{b} \,\in\, \mathbb{R}^3}{\text{minimize}} \sum_{i=1}^{n} \left\| \boldsymbol{\tau}_i^T \boldsymbol{a} - y_i \boldsymbol{\tau}_i^T \boldsymbol{b} \right\|^2 \;\; s.t. \;\; \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b} < 0, \tag{5}$$

where $\boldsymbol{\tau}_i = \boldsymbol{\tau}(t_i)$. Note that although the unknowns are 6, the DoF of the problem are 5 due to the negative discriminant constraint. To solve it, we consider the Lagrangian function,

$$\mathcal{L} = \sum_{i=1}^{n} \left\| \boldsymbol{\tau}_i^T \boldsymbol{a} - y_i \boldsymbol{\tau}_i^T \boldsymbol{b} \right\|^2 - \lambda \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b}, \tag{6}$$

where $\lambda \geq 0$ is a Lagrange multiplier. Using matrices, the Lagrangian can be written as

$$\mathcal{L} = \boldsymbol{a}^T \boldsymbol{A} \boldsymbol{a} - 2 \boldsymbol{a}^T \boldsymbol{B} \boldsymbol{b} + \boldsymbol{b}^T \boldsymbol{C} \boldsymbol{b} - \lambda \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b}, \tag{7}$$

where

$$\boldsymbol{A} = \sum_{i=1}^{n} \boldsymbol{\tau}_i \boldsymbol{\tau}_i^T, \quad \boldsymbol{B} = \sum_{i=1}^{n} y_i \, \boldsymbol{\tau}_i \boldsymbol{\tau}_i^T, \quad \boldsymbol{C} = \sum_{i=1}^{n} y_i^2 \, \boldsymbol{\tau}_i \boldsymbol{\tau}_i^T.$$

Setting the partial derivative of $\mathcal{L}$ with respect to $\mathbf{a}$ to zero yields a relationship between $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}} = 2 \boldsymbol{A} \boldsymbol{a} - 2 \boldsymbol{B} \boldsymbol{b} = 0 \iff \boldsymbol{a} = \boldsymbol{A}^{-1} \boldsymbol{B} \boldsymbol{b}. \tag{8}$$

Similarly, setting the partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{b}$ to zero and substituting from (8) yields a generalized eigenvalue problem which can be readily turned into a standard one:

$$\left(\boldsymbol{C} - \boldsymbol{B} \boldsymbol{A}^{-1} \boldsymbol{B}\right) \boldsymbol{b} = \lambda \boldsymbol{K} \boldsymbol{b} \iff \boldsymbol{K}^{-1} \left(\boldsymbol{C} - \boldsymbol{B} \boldsymbol{A}^{-1} \boldsymbol{B}\right) \boldsymbol{b} = \lambda \boldsymbol{b} \tag{9}$$

The solution(s) for $\boldsymbol{b}$ will be the eigenvectors of matrix $\boldsymbol{K}^{-1} \left(\boldsymbol{C} - \boldsymbol{B} \boldsymbol{A}^{-1} \boldsymbol{B}\right)$ that correspond to positive eigenvalues. The latter can be computed analytically as the roots of the cubic characteristic polynomial. The solution for $\boldsymbol{a}$ is then recovered from $\boldsymbol{b}$ via (8). The inverse of matrix $\boldsymbol{A}$ involved in eqs. (8) and (9) is computed with the aid of the LDLT (i.e., square root free Cholesky) decomposition [30].

2) Approximation with fixed points: We now extend our analysis by considering a more general case of regression in which a non-empty subset of the data points is interpolated instead of simply being approximated by f. Note that using hard constraints is a sensible design requirement in pose regression, if we consider that features are typically tracked from the most recent past frame(s) to the current one and we therefore want to make sure that the curve does not deviate from the latest pose(s) under the influence of older estimates. Let $\mathcal{C} = \left\{ j_1, \ldots j_{|\mathcal{C}|} \right\}$ be the set of the indexes of the fixed data points. We augment the problem of (5) with linear equality constraints on the fixed points:

$$\begin{aligned} &\underset{\boldsymbol{a},\, \boldsymbol{b} \,\in\, \mathbb{R}^3}{\text{minimize}} \sum_{i=1}^{n} \left\| \boldsymbol{\tau}_i^T \boldsymbol{a} - y_i \boldsymbol{\tau}_i^T \boldsymbol{b} \right\|^2 \\ &s.t. \;\; \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b} < 0, \;\; \boldsymbol{\tau}_j^T \boldsymbol{a} - y_j \boldsymbol{\tau}_j^T \boldsymbol{b} = 0 \;\; \forall \, j \in \mathcal{C}. \end{aligned} \tag{10}$$

Clearly, imposing hard data constraints is meaningful only when $n > 5$ (i.e., the problem is overdetermined), in which case the cardinality $|\mathcal{C}|$ of $\mathcal{C}$ can be at most 4.

To solve the problem of (10), we resort once again to the corresponding Langrangian function, this time including the terms associated with the fixed points indexed by $\mathcal{C}$:

$$\mathcal{L} = \boldsymbol{a}^T \boldsymbol{A} \boldsymbol{a} - 2 \boldsymbol{a}^T \boldsymbol{B} \boldsymbol{b} + \boldsymbol{b}^T \boldsymbol{C} \boldsymbol{b} - \lambda \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b} + \mathbf{1}_{|\mathcal{C}|}^T \boldsymbol{M} \left(\boldsymbol{D} \boldsymbol{a} - \boldsymbol{E} \boldsymbol{b}\right). \tag{11}$$

Vector $\mathbf{1}_{|\mathcal{C}|}$ is of length $|\mathcal{C}|$ with all entries equal to 1, and

$$\begin{aligned} \boldsymbol{D} &= \begin{bmatrix} \boldsymbol{\tau}_{j_1} & \ldots & \boldsymbol{\tau}_{j_{|\mathcal{C}|}} \end{bmatrix}^T, \; \boldsymbol{M} = \operatorname{diag}\left\{ \mu_1, \ldots, \mu_{|\mathcal{C}|} \right\}, \\ \boldsymbol{E} &= \begin{bmatrix} y_{j_1} \boldsymbol{\tau}_{j_1} & \ldots & y_{j_{|\mathcal{C}|}} \boldsymbol{\tau}_{j_{|\mathcal{C}|}} \end{bmatrix}^T, \end{aligned} \tag{12}$$

where $\mu_1, \ldots, \mu_{|\mathcal{C}|}$, are the Lagrange multipliers corresponding to the fixed points. From the above, it follows that $\operatorname{rank}(\boldsymbol{D}) = \min\{3, |\mathcal{C}|\}$. We can therefore express $\mathbf{a}$ in terms of the null and row space of $\boldsymbol{D}$ as follows,

$$\boldsymbol{a} = \boldsymbol{N} \boldsymbol{a}' + \boldsymbol{H} \boldsymbol{a}'', \tag{13}$$

where $\boldsymbol{N} \in \mathbb{R}^{3 \times (3 - \min\{3, |\mathcal{C}|\})}$ and $\boldsymbol{H} \in \mathbb{R}^{3 \times \min\{3, |\mathcal{C}|\}}$ contain bases of the null and row space of $\boldsymbol{D}$ in column arrangement and $\boldsymbol{a}' \in \mathbb{R}^{3 - \min\{3, |\mathcal{C}|\}}$, $\boldsymbol{a}'' \in \mathbb{R}^{\min\{3, |\mathcal{C}|\}}$. Clearly, if $|\mathcal{C}| \geq 3$ it follows that $\boldsymbol{N} = \mathbf{0}$. Substituting eq. (13) into the hard constraint equality, $\boldsymbol{D}\boldsymbol{a} = \boldsymbol{E}\boldsymbol{b}$, we get

$$\begin{aligned} \boldsymbol{D} \left(\boldsymbol{N} \boldsymbol{a}' + \boldsymbol{H} \boldsymbol{a}''\right) = \boldsymbol{E} \boldsymbol{b} \iff \boldsymbol{D} \boldsymbol{H} \boldsymbol{a}'' = \boldsymbol{E} \boldsymbol{b} \iff \\ \boldsymbol{a}'' = \left(\boldsymbol{H}^T \boldsymbol{D}^T \boldsymbol{D} \boldsymbol{H}\right)^{-1} \left(\boldsymbol{D} \boldsymbol{H}\right)^T \boldsymbol{E} \boldsymbol{b} = \boldsymbol{F} \boldsymbol{b}. \end{aligned} \tag{14}$$

For brevity, let $\boldsymbol{F} = \left(\boldsymbol{H}^T \boldsymbol{D}^T \boldsymbol{D} \boldsymbol{H}\right)^{-1} \left(\boldsymbol{D} \boldsymbol{H}\right)^T \boldsymbol{E}$. By substituting (13) and (14) into (11), the term associated with the fixed points vanishes (since the constraint is satisfied) and the Lagrangian is expressed in terms of $\boldsymbol{a}'$ and $\boldsymbol{b}$ as

$$\mathcal{L} = \boldsymbol{a}'^T \overline{\boldsymbol{A}} \boldsymbol{a}' - 2 \boldsymbol{a}'^T \overline{\boldsymbol{B}} \boldsymbol{b} + \boldsymbol{b}^T \overline{\boldsymbol{C}} \boldsymbol{b} - \lambda \, \boldsymbol{b}^T \boldsymbol{K} \boldsymbol{b}, \tag{15}$$

where

$$\begin{aligned} &\overline{\boldsymbol{A}} = \boldsymbol{N}^T \boldsymbol{A} \boldsymbol{N}, \;\; \overline{\boldsymbol{B}} = \boldsymbol{N}^T \left(\boldsymbol{B} - \boldsymbol{A} \boldsymbol{H} \boldsymbol{F}\right), \\ &\overline{\boldsymbol{C}} = \left(\boldsymbol{H} \boldsymbol{F}\right)^T \boldsymbol{A} \boldsymbol{H} \boldsymbol{F} + \boldsymbol{C} - 2 \left(\boldsymbol{H} \boldsymbol{F}\right)^T \boldsymbol{B}. \end{aligned} \tag{16}$$

Having eliminated the equality constraints, the Lagrangian of eq. (15) is an expression similar to (7) in section III-B. In quite the same way, $\boldsymbol{a}'$ can be eliminated from the first order conditions, as follows

$$\boldsymbol{a}' = \overline{\boldsymbol{A}}^{-1}\overline{\boldsymbol{B}}\,\boldsymbol{b}. \tag{17}$$

This eventually leads to yet another generalized eigenvalue problem, which is turned into a standard eigenvalue problem by multiplying with $\boldsymbol{K}^{-1}$ from the left

$$\begin{aligned} \left(\frac{\overline{\boldsymbol{C}}+\overline{\boldsymbol{C}}^T}{2} - \overline{\boldsymbol{B}}^T\overline{\boldsymbol{A}}^{-1}\overline{\boldsymbol{B}}\right)\boldsymbol{b} &= \lambda\boldsymbol{K}\boldsymbol{b} \iff \\ \boldsymbol{K}^{-1}\left(\frac{\overline{\boldsymbol{C}}+\overline{\boldsymbol{C}}^T}{2} - \overline{\boldsymbol{B}}^T\overline{\boldsymbol{A}}^{-1}\overline{\boldsymbol{B}}\right)\boldsymbol{b} &= \lambda\boldsymbol{b}, \end{aligned}$$

and the solution(s) for $\boldsymbol{b}$ will be the eigenvectors that correspond to positive values of λ. The solution for $\boldsymbol{a}$ can be recovered from $\boldsymbol{b}$ using eqs. (17), (16) and (13).

IV. Experiments

This section reports results from the application of the proposed method to data originating from our feature-based stereo vSLAM system, whose tracking back-end is a variant of the point tracker of PTAM [7] and ORB-SLAM [8]. More specifically, three small inter-frame pixel displacement sequences captured at 30Hz in an indoor industrial environment were employed, collectively exceeding 24K frames in total. The sequences were processed by our vSLAM system which operated as normal, performing its loop closures and graph optimizations. The camera poses estimated for each sequence are visualized in Figure 2.

A. Predicted vs optimized pose

Using a moving time window of length $n = 7$ for each of the 6 pose parameters (3 for the MRP + 3 for location), the proposed method predicted the poses for all frames of the three sequences. Poses were also predicted with Matlab implementations of the rational method of [21], polynomials of degrees 3 and 4, Gaussian processes [15] and cubic splines [31]. Finally, the predicted poses were compared against those estimated by our vSLAM system, treating the latter as ground truth.

To quantify the error for a predicted pose, the following rotational and positional errors were employed. Given a camera pose estimated by vSLAM (composed of a rotation matrix $\mathbf{R}_\mathrm{s}$ and position vector $\mathbf{p}_\mathrm{s}$), the error for an estimated rotation $\mathbf{R}_\mathrm{e}$ is the angle of rotation about a unit vector that transfers $\mathbf{R}_\mathrm{e}$ to $\mathbf{R}_\mathrm{s}$, computed as $\arccos((\mathrm{trace}(\mathbf{R}_\mathrm{s}\mathbf{R}_\mathrm{e}^\mathrm{T}) - 1)/2)$. The error for a predicted position $\mathbf{p}_\mathrm{e}$ is simply the Euclidean norm of the difference of the position parts, i.e. $\|\mathbf{p}_\mathrm{s} - \mathbf{p}_\mathrm{e}\|$.

Figure 3 shows the rotation errors pertaining to the estimates computed by each interpolation method for all three sequences. The proposed method consistently outperforms the other ones with that of [21] being the second best. Positional errors are shown in Fig. 4, where the proposed method again performs best.

B. Successfully matched features

The number of matches in vSLAM is clearly a strong quality indicator of tracking performance. We therefore generated histograms of the number of features tracked during the primary (frame-to-frame) and map (map-to-frame) tracking phases over the first 5000 consecutive frames of the second sequence in Fig. 2, which was chosen due to its increased twists. Figure 5 overlays the histogram pertaining to our method during primary tracking over these of competing methods. Similar histograms are shown in Figure 6 for map tracking. Regarding the choice of competing prediction methods, we used those we considered in Sec. IV-A, excluding [21] as well as polynomial splines due to the fact that there are no C++ implementations readily available for them.

Each histogram in Figures 5 and 6 illustrates the distribution of feature matches from rational function pose prediction against a rival method indicated in the legend. This is effectively a distribution of feature matches across all types of conditions related to camera motion. This means that benign vSLAM conditions (e.g, slow and stable camera motion, absence of occlusions, rich texture, etc.) will favor the less effective methods in these results. Yet, even with the presence of such bias, we observe that rational pose prediction has always a higher average number of 20 to 50 matches without a single exception in these plots. Overall, rational pose prediction results to an increment in the frequency of matched features numbers between 200 and 400 of up to 50 in each plot. Furthermore, for features matches above 250, rational pose prediction yields almost certainly higher frequencies as indicated by the green histogram being always higher than any of the other methods in the > 250 matches region.

V. Conclusions

Compared to polynomial splines, rational functions are known to be smoother and less oscillatory data-fitting models. However, they give rise to hard optimization problems due to the potential presence of zeros in their denominator. This work has suggested a method for stable rational regression of 6D pose in visual SLAM, aiming at more efficient feature tracking without the employment of a motion model prior or IMU readings. Comprehensive experiments performed with real-world data originating from a feature-based vSLAM pipeline have demonstrated the effectiveness of the approach. A C++ implementation is publicly available at `https://github.com/terzakig/RRP`.

Acknowledgment

This work was partially funded by the EU's H2020 & Horizon Europe research and innovation programmes under GAs 101017151 (FELICE) and 101120990 (SOPRANO), respectively.

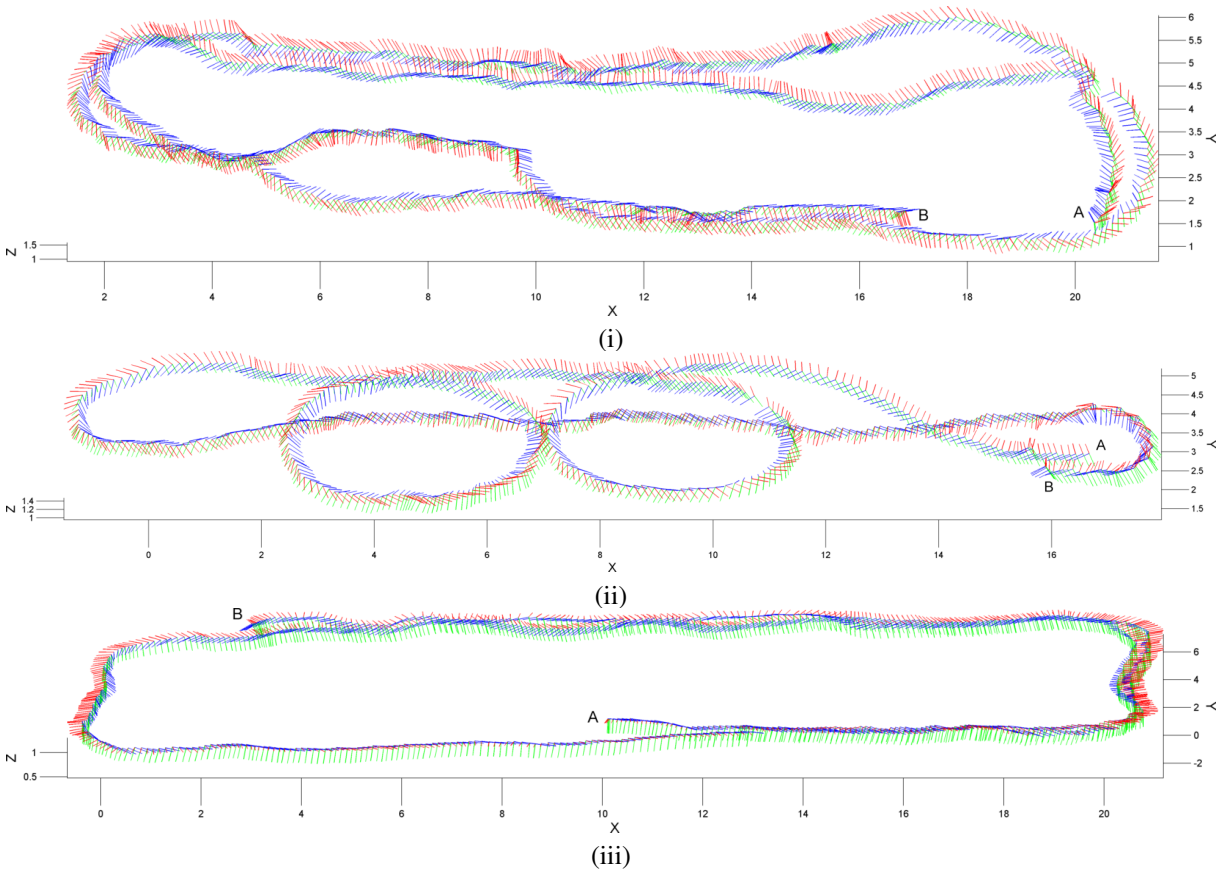

Fig. 2. 3D plots of the local camera axes that correspond to the employed image sequences. Local axes are plotted as moving trihedrals using the common computer vision convention (X points right, Y down and Z into the scene). The axes are colored using red for X, green for Y and blue for Z. The camera starts at the points labeled with As and terminates at Bs; axis units are in meters. Sequence (i) consists of 9262 frames, (ii) of 6426 and (iii) of 8491. To reduce clutter, only one every 10 camera poses is shown.

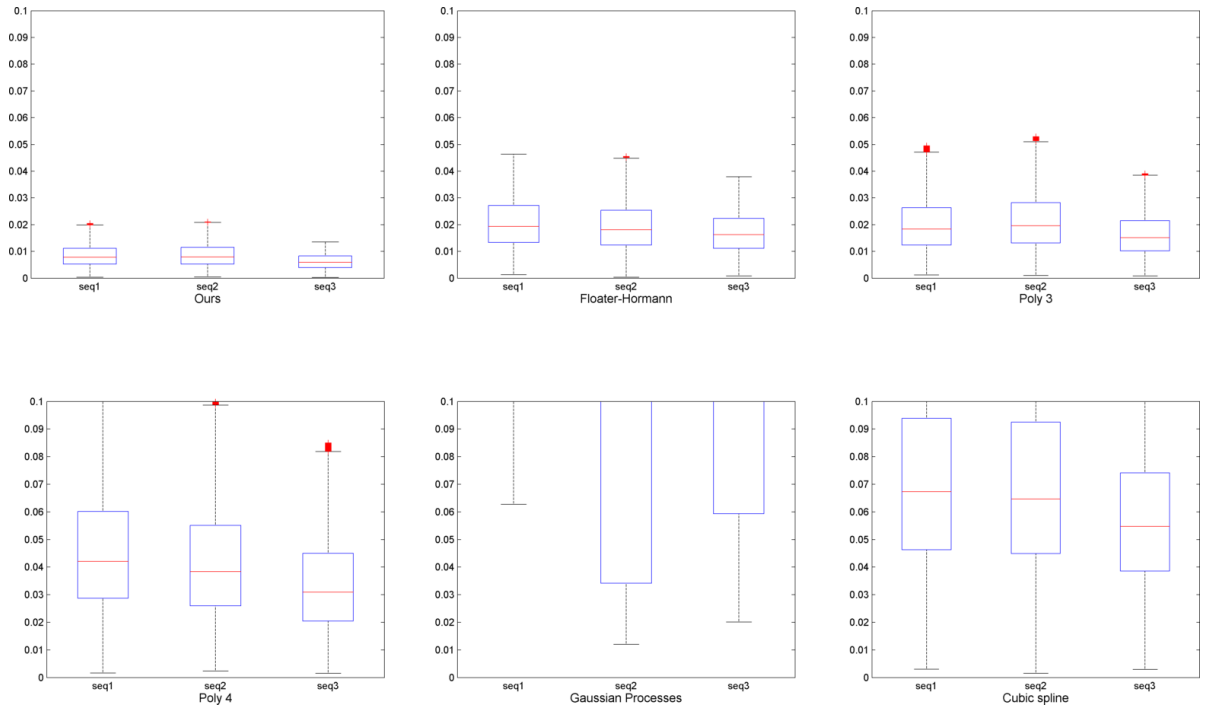

Fig. 3. Box plots of the rotational errors (in rad) for the interpolants computed by the six methods being compared on all poses predicted for the three sequences of Fig. 2. Observe that all Y axes have been clipped at 0.1 rad.

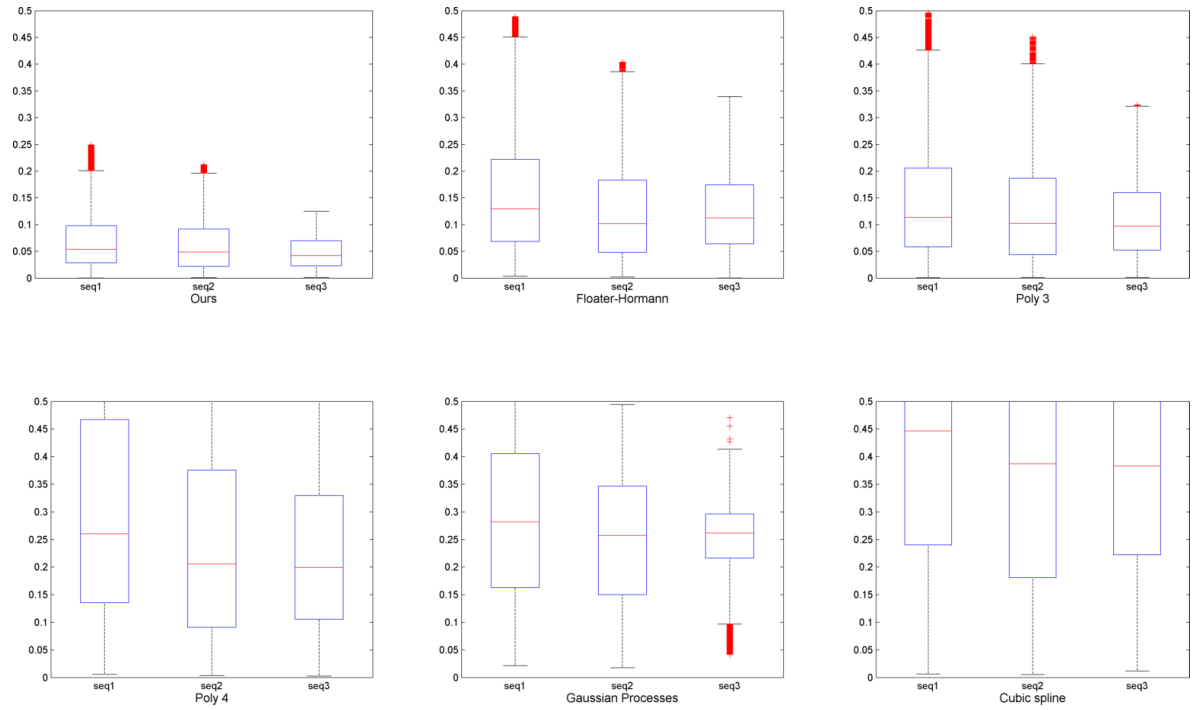

Fig. 4. Box plots of the positional errors (in m) for the interpolants computed by the six methods being compared on all poses predicted for the three sequences of Fig. 2. The Y axes have been clipped at 0.5 m.

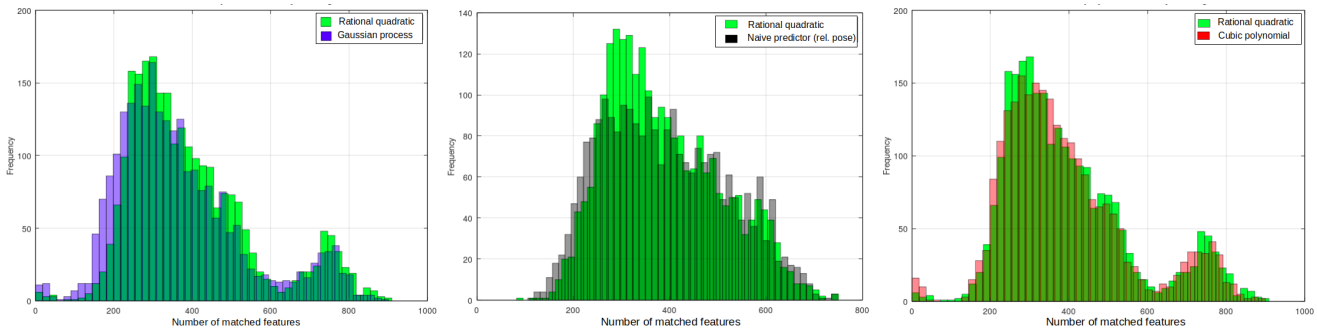

Fig. 5. Histograms of feature matches in primary (frame-to-frame) tracking over the first 5000 vSLAM tracker invocations for the sequence of Fig. 2(ii).

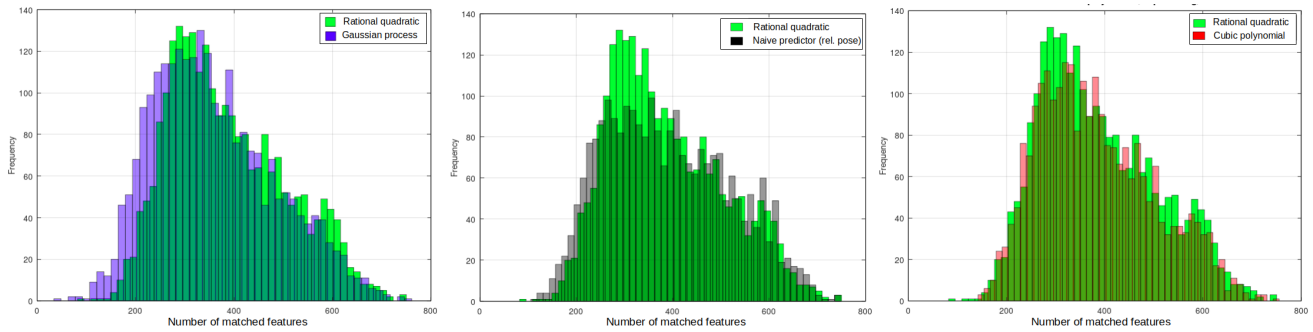

Fig. 6. Histograms of feature matches during map-to-frame tracking in the first 5000 vSLAM tracker invocations for the sequence of Fig. 2(ii).

REFERENCES

[1] F. Ott, T. Feigl, C. Loffler, and C. Mutschler, "ViPR: visual-odometry-aided pose regression for 6DoF camera localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 42–43.

[2] H. Xiu, Y. Liang, H. Zeng, Q. Li, H. Liu, B. Fan, and C. Li, "Robust self-supervised monocular visual odometry based on prediction-update

pose estimation network," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105481, 2022.

[3] R. Li, S. Wang, and D. Gu, "Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities," *Cognitive Computation*, vol. 10, pp. 875–889, 2018.

[4] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3302–3312.

[5] H. Ovrén and P.-E. Forssén, "Trajectory representation and landmark projection for continuous-time structure from motion," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 686–701, 2019.

[6] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2088–2095.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[9] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[11] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[12] M. Persson, G. Häger, H. Ovrén, and P.-E. Forssén, "Practical pose trajectory splines with explicit regularization," in *International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 156–165.

[13] T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," in *Proceedings of Robotics: Science and Systems*, vol. 10, Berkeley, USA, 2014, pp. 1–10.

[14] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, "Continuous-time batch trajectory estimation using temporal basis functions," *The International Journal of Robotics Research*, vol. 34, no. 14, pp. 1688–1710, 2015.

[15] S. Anderson and T. D. Barfoot, "Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 157–164.

[16] J. Dong, M. Mukadam, B. Boots, and F. Dellaert, "Sparse Gaussian processes on matrix lie groups: A unified framework for optimizing continuous-time trajectories," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6497–6504.

[17] B. Fornberg and J. Zuev, "The Runge phenomenon and spatially variable shape parameters in RBF interpolation," *Computers & Mathematics with Applications*, vol. 54, no. 3, pp. 379–398, 2007.

[18] G. Dahlquist and Å. Björck, "Equidistant interpolation and the Runge phenomenon," *Numerical Methods*, pp. 101–103, 1974.

[19] P. P. Petrushev and V. A. Popov, *Rational approximation of real functions*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2011, no. 28.

[20] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, "The AAA algorithm for rational approximation," *SIAM Journal on Scientific Computing*, vol. 40, no. 3, pp. A1494–A1522, 2018.

[21] M. S. Floater and K. Hormann, "Barycentric rational interpolation with no poles and high rates of approximation," *Numerische Mathematik*, vol. 107, no. 2, pp. 315–331, Aug. 2007.

[22] S.-I. Filip, Y. Nakatsukasa, L. N. Trefethen, and B. Beckermann, "Rational minimax approximation via adaptive barycentric representations," *SIAM Journal on Scientific Computing*, vol. 40, no. 4, pp. A2427–A2455, 2018.

[23] Y. Nakatsukasa and L. N. Trefethen, "An algorithm for real and complex rational minimax approximation," *SIAM Journal on Scientific Computing*, vol. 42, no. 5, pp. A3157–A3179, 2020.

[24] G. Lentaris, I. Stamoulias, D. Soudris, and M. Lourakis, "HW/SW codesign and FPGA acceleration of visual odometry algorithms for rover navigation on Mars," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1563–1577, 2016.

[25] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 2014, pp. 834–849.

[26] J. L. Crassidis and F. L. Markley, "Attitude estimation using modified Rodrigues parameters," in *Flight Mechanics/Estimation Theory Symposium*, 1996.

[27] G. Terzakis, M. Lourakis, and D. Ait-Boudaoud, "Modified Rodrigues parameters: an efficient representation of orientation in 3D vision and graphics," *Journal of Mathematical Imaging and Vision*, vol. 60, pp. 422–442, 2018.

[28] M. Lourakis and G. Terzakis, "A globally optimal method for the PnP problem with MRP rotation parameterization," in *International Conference on Pattern Recognition (ICPR)*, 2021, pp. 3058–3063.

[29] M. D. Shuster, "A survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.

[30] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. USA: Johns Hopkins University Press, 1996.

[31] R. Bartels, J. Beatty, and B. Barsky, "Hermite and cubic spline interpolation," in *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, ser. Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, 1998, ch. 3, pp. 9–17.