Practical Pose Trajectory Splines With Explicit Regularization

Mikael Persson[1] Gustav Häger[1] Hannes Ovrén[2] Per-Erik Forssén[1]

Abstract

We investigate spline-based continuous-time pose trajectory estimation using non-linear explicit motion priors. Current regularization priors either linearize the orientation, rely on the implicit regularization obtained from the used spline basis function, or use sampling based regularization schemes. The latter is a special case of a Riemann sum approximation, and we demonstrate when and why this can fail, and propose a way to avoid these issues. In addition we provide a number of novel practically useful theoretical contributions, including requirements on knot spacing for orientation splines, new basis functions for constant velocity extrapolation, and a generalization of the popular P-Spline penalty to orientation. We analyze the properties of the proposed approach using synthetic data. We validate our system using the standard task of visual-inertial calibration, and apply it to stereo visual odometry where we demonstrate real-time performance on KITTI.

1. Introduction

State estimation problems in computer vision are usually modeled by estimating the sensor pose only at a discrete set of times (for example, one per camera frame). This works well for data which is itself discretely sampled e.g. global shutter camera images in reconstruction from collections of images [1, 34] and low framerate video [31]. Many sensors provide data more akin to a continuous stream than discrete samples, e.g. high framerate video, rolling shutter images, spinning lidars, event cameras, and inertial sensors. These are better represented by continuous-time pose trajectories [12]. Such *pose splines* allow the convenient incorporation of derivative measurements, such as from an inertial measurement unit (IMU), as predictions of the analytic trajectory derivatives. High frame-rate video Visual Odometry (VO) illustrates another advantage. Standard discrete bundle adjustment approaches scale cubically with the number of frames, and as a result, often necessitate key framing i.e. tossing information. Continuous pose trajectories with local control instead scale cubically with the number of knots, which can be much lower than the frame rate [21].

Basis splines are a natural fit for continuous-time poses, and were introduced for this purpose by Furgale et al. [12]. They have since been refined by many follow-up papers e.g. [24, 14, 4, 28, 23, 25, 26, 37]. While the continuous-time approach provides clear benefits over the discrete-time approach, it has not yet been widely adopted by the computer vision community, outside niche topics such as rolling shutter structure from motion, and lidar fusion.

In the pose spline formulation, motion models are incorporated as a prior term in the cost, and need thus not be linearized. We argue that they should instead approximated using a Riemann sum, and propose a lower bound for the sampling rate (i.e. the inverse sub-interval width) based on observability, and frequency analysis.

We provide several practical improvements to the theoretical framework of pose trajectory estimation:
1) We generalize the unit quaternion spline (KKS) by Kim, Kim and Shin [20], by extending it to infinity using constant angular velocity. We do this by modifying the basis functions at the beginning and end in such a way that the spline locality and continuity is maintained.
2) We provide a simpler way to derive and compute integrals of spline derivative product functions.
3) We provide an upper bound on the angular velocity $\omega(t)$, for the KKS spline, which in turn has important implications for the sampling and knot distance selection.

These contributions, together with the provided optimization framework code [29] should make it easier to use pose splines for various applications in geometric vision and robotics.

2. Related Work

Here we describe related work on continuous-time trajectories, and on regularization strategies for pose splines.

An excellent overview of continuous-time pose trajectories is given in Furgale et al. [14]. The methods discussed by Furgale mostly fall into two categories: They are either KKS splines [20], or equivalent to a straightforward approximation, such as a linearization of a KKS spline. For instance, the popular SLERP construction [36] on pairs of consecutive control points corresponds to a first order KKS spline, and splines over non-quaternion rotation representa-

[1]first.last@liu.se, Linköping University, Computer Vision Laboratory
[2]first.last@foi.se, Swedish Defence Research Agency

tions are either equivalent to, or linearizations of the KKS spline. Many of the more recent papers also use the KKS construct [28, 23, 25, 26, 37].

The extension of classic Bayesian discrete filtering approaches to continuous states i.e. Gaussian Processes (GP) by Barfoot et al. [5] and especially pose trajectory GP frameworks [2, 9] are relevant if the full posterior and not just a point estimate is needed. These frameworks are similar to a zeroth or first degree version of [14] using Gaussian observer functions. In principle GPs are a powerful approach, but in practice, the price is high. In order to achieve feasible computation, the GP pose frameworks linearize the rotation, and since they lose local control, short optimization windows are generally used. This work instead focuses on point estimates using KKS pose splines with local control, and on-manifold rotation representation.

The range of motions a spline can express is constrained by the choice of basis functions, their degree and the knot spacing. This has previously been used as a regularizer although it might not be explicitly stated as such [20, 19, 23, 6]. Bibby et al. [6] discuss the connection between spline parameters and the constraints on dynamics. Anderson et al. [3], and Ovrén et al. [25], investigated how to select a knot spacing so as to ensure the spline could represent the necessary motions. Oth et al. [24] instead use a variable knot spacing. They find the spacings by successively inserting control points during optimization at any interval where residuals are not normally distributed, or moving knots further apart until the predicted observation error variances match the measurement model. Such non-causal methods work well in batch, e.g. for offline calibration [13, 24], but are ill-suited to causal prediction or real time applications.

Incorporation of inertial measurements reduces the need for explicit regularization, and they are thus sometimes also referred to as regularizing measurements [27, 25]. Depending on what other type of measurements are available, there is however a risk of overfitting to one modality. In [25] this problem is addressed by adaptively weighting the inertial measurements based on the motion energy present in the inertial data, and the chosen spline knot spacing.

A common form of trajectory regularization is a cost term derived from a motion model. Motion models usually result in cost terms of integral type, e.g. a constant angular velocity $\boldsymbol{\omega}(t)$ model with Gaussian process noise results in the cost term $\int_{\mathbb{R}} \frac{\partial \boldsymbol{\omega}(t)}{\partial t}^2 dt$. However, the non-linear nature of rotation means that in all but the simplest cases these cannot be used to formulate analytic maximum aposteriori (MAP) estimation problems without approximation.

One approach is to approximate the model formulation. Furgale et al. [14] use an $\mathbb{R}^6$ spline state to represent the pose trajectory. The state is regularized by the integral of the second derivative of the spline. With regards to the translation, this has an intuitive interpretation; it penalizes the linear acceleration. However, with regards to the rotation it does not, as the angular acceleration is a non-linear function of the parameters. This approach also linearizes expressions involving the rotation. This requires that both true orientation, and estimate at every iteration, must lie within a small region. In the discrete case, the linearization is typically re-centered as needed, but doing so in the continuous case is not possible without losing either continuity or linearity.

Cost terms of integral form can also be approximated by sampling the integrand [28, 21, 18, 4]. In [28, 21] a single sample per knot is used, whereas [39] uses one sample per observation. The former are Riemann approximations, and the latter could be made into such by weighting according to sampling frequency. The Riemann approximation is well known, but there are critical differences between approximating an integral to estimate the value, and approximating a cost term, as will be discussed in Section 5.5. Furgale and Anderson [14, 4] observe issues caused by the state becoming unobservable as knot distance decreases, and note the need to lower knot distances to account for high frequency dynamics. This class of methods along with several other approaches [28, 21, 39, 18] would have these issues as well. However, their experiments do not show this due to the use of relatively dense measurements, though e.g. Huang et al. [18] do note that they were unable to represent a high frequency signal in one experiment. In this work we establish a lower bound for the sampling rate, which is notably above the one/knot used by [28, 21], and show how it is related to the minimum degree and maximum knot distances given signal properties.

Huang et al. [18] consider different ways to regularize trajectories using a forward-only vehicle model and investigate the alternatives with regards to noise sensitivity. Their variant SSBARv is an orientation spline estimated from visual measurements where the regularization is applied as an L^2 penalty on the deviation from forward velocity at the frames. The open issues with the SSBARv method, i.e. the uneven weighting over time, what the weight should be, and the knot distance required for an orientation spline to express a given frequency, are answered by this work.

3. Basis Splines

Here we briefly introduce basis splines and give a few novel and/or practical extensions. For a comprehensive introduction to basis splines and quaternion splines in general, see [38, 8, 20]. We use the Δ operator, which is defined by $\Delta \mathbf{a}_i = \mathbf{a}_i - \mathbf{a}_{i-1}$ and $\Delta^n \mathbf{a}_i = \Delta^{n-1}(\Delta \mathbf{a}_i)$ for $\mathbf{a}_i \in \mathbb{R}^N$. We use dots to denote time derivatives, e.g. $\ddot{\mathbf{a}}(t)$ is $\frac{\partial^2 \mathbf{a}(t)}{\partial t^2}$. We will describe uniform splines with knot distance $\Delta t_i = 1$, as the generalizations are trivial, but verbose. Further we choose the knots t_i as i, for $i \in \mathbb{Z}$. Now

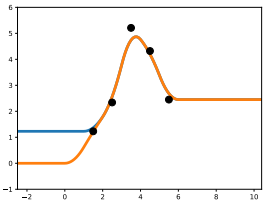

Figure 1. Backwards extrapolation from (6) (orange) and (7) (blue) for $n = 2$. Black dots are control points, $(i + (n + 1)/2, a_i)$.

our uniform $\mathbb{R}^N$ B-splines of degree n, with control points $\mathbf{a}_i$ are defined as in e.g. [38]:

$$\mathbf{s}_n(t) = \sum_i \mathbf{a}_i b_n(t - i) \tag{1}$$

where $b_n(t)$ is defined according to:

$$b_0(t) = \begin{cases} 1 & \text{if } t \in [0, 1) \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

$$b_n(t) = (b_0 \circledast b_{n-1})(t) = (b_0 \circledast b_0 \ldots \circledast b_0)(t) \tag{3}$$

where $(f \circledast g)(t) = \int f(\tau) g(t-\tau) d\tau$ denotes a convolution. We will refer to this as the *local form*, and it is equivalent to the *cumulative form*, as defined by e.g. Kim et al. [20]:

$$c_n(t) = \sum_{i=0}^{\infty} b_n(t-i) = \begin{cases} \sum_{i=0}^{n-1} b_n(t-i) & \text{if } t < n \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Now (1), for a control point interval $[B, E]$, becomes:

$$\mathbf{s}_n(t) = \sum_{i=B}^{E} \mathbf{a}_i b_n(t - i) = \tag{5}$$

$$\mathbf{a}_B c_n(t - B) + \sum_{i=B+1}^{E} \Delta\mathbf{a}_i c_n(t - i) \tag{6}$$

where $\Delta\mathbf{a}_i = \mathbf{a}_i - \mathbf{a}_{i-1}$. By omitting the basis function on the first control point, we obtain:

$$\mathbf{s}_n(t) = \mathbf{a}_B + \sum_{i=B+1}^{E} \Delta\mathbf{a}_i c_n(t - i) \tag{7}$$

This expression extrapolates backwards using the first knot instead of zero, see figure 1.

By using an adaptive start index, $f = \lfloor t \rfloor - n - 1$, we finally obtain the expression used in practice:

$$\mathbf{s}_n(t) = \mathbf{a}_f + \sum_{i=f+1}^{\lfloor t \rfloor} \Delta\mathbf{a}_i c_n(t - i)\,. \tag{8}$$

3.1. Differential Relations

B-splines are straightforward to differentiate and integrate, and the following relations are useful [38, 20]:

$$\frac{\partial}{\partial t} \sum_i \mathbf{a}_i b_n(t - i) = \sum_i \Delta\mathbf{a}_i b_{n-1}(t - i) \tag{9}$$

$$\frac{\partial c_n(t)}{\partial t} = b_{n-1}(t) \tag{10}$$

We also note that:

$$\frac{\partial}{\partial t} b_0(t) = \delta(t) - \delta(t - 1) \text{ where} \tag{11}$$

$$\int_a^b \delta(t - i) f(t) dt = f(i) \text{ if } i \in [a, b]\,. \tag{12}$$

We propose to use the often overlooked property (11) to significantly simplify integrals of simple functions of splines. For instance, given a second degree spline $s_2(t)$ with white Gaussian accelerations, the regularizing cost can be derived using partial integration and (9)-(11):

$$\int_a^b \dddot{s}_2^{\,2} dt = [\dddot{s}_2 \ddot{s}_2]_a^b - \int_a^b \dddot{s}_2 \ddot{s}_2 dt = \tag{13}$$

$$[\dddot{s}_2 \ddot{s}_2]_a^b - \sum_i \Delta^2\mathbf{a}_i \int_a^b \dddot{s}_2(\delta(t-i) - \delta(t-i-1)) dt = \tag{14}$$

$$[\dddot{s}_2 \ddot{s}_2]_a^b - \sum_{i \in [a,b]} \dddot{s}_2(i) \Delta^3\mathbf{a}_i \tag{15}$$

This approach is significantly simpler than the common matrix formulation approach taken by e.g. Furgale et al. [14], in Eqn 37 to 59 for the same problem. Our approach generalizes to a broad category of spline differential functions, and has to the best of our knowledge been overlooked until now. Additional examples are found in the supplementary material.

3.2. KKS Quaternion Spline

The KKS(Kim, Kim and Shin) quaternion spline [20] is a generalization of (6). The control points $\mathbf{k}_i$ are unit quaternions and the operations are replaced by their rotation equivalents i.e. addition with quaternion multiplication, and control point weighting by exponentiation. The Δ operator for unit quaternions thus becomes $\Delta\mathbf{k}_i = \mathbf{k}_{i-1}^* \mathbf{k}_i$. Using $\mathbf{w}_i = \Delta\mathbf{k}_i$ for brevity the KKS is defined as:

$$\mathbf{q}_n(t) = \mathbf{k}_F^{c_n(t-F)} \prod_{i=F}^{\infty} \exp(\log(\mathbf{w}_i) c_n(t - i))\,. \tag{16}$$

Where $\exp()$ and $\log()$ map between unit quaternions and their tangent space. By instead using (7) and $\mathbf{q}^a =$

$\exp(\log(\mathbf{q})a)$ we obtain a simpler definition of the orientation spline as:

$$\mathbf{q}_n(t) = \mathbf{k}_F \prod_{i=F}^{\infty} \mathbf{w}_i^{c_n(t-i)} = \quad \mathbf{k}_f \prod_{i=f+1}^{\lfloor t \rfloor} \mathbf{w}_i^{c_n(t-i)} \tag{17}$$

for $f = \lfloor t \rfloor - n - 1$ as in (8). This formulation also extrapolates backwards towards $\mathbf{k}_F$ rather than identity. Using helper variables τ, W, T we find the recursive form of (17):

$$\boldsymbol{\tau}_{n,\,i}(t) = \boldsymbol{\tau}_{n,\,i-1}(t)\mathbf{w}_i^{c_n(t-i)}, \quad \boldsymbol{\tau}_{n,\,f} = \mathbf{k}_f \tag{18}$$

$$\mathbf{q}_n(t) = \boldsymbol{\tau}_{n,\,\lfloor t \rfloor}(t) \tag{19}$$

This simplifies computing derivatives [20]:

$$\mathbf{W}_i(t) = \mathbf{w}_i^{c_n(t-i)}, \quad \dot{\mathbf{W}}_0(t) = 0 \tag{20}$$

$$\dot{\mathbf{W}}_i(t) = b_{n-1}(t-i)\log(\mathbf{w}_i)\mathbf{W}_i(t) \tag{21}$$

$$\dot{\boldsymbol{\tau}}_{n,\,0}(t) = 0 \tag{22}$$

$$\dot{\boldsymbol{\tau}}_{n,i}(t) = \dot{\boldsymbol{\tau}}_{n,\,i-1}(t)\mathbf{W}_i(t) + \mathbf{q}_n(t)\dot{\mathbf{W}}_i(t) \tag{23}$$

which gives the angular velocity $\boldsymbol{\omega}(t)$ in rad/s as

$$\mathbf{T}_{n,i}(t) = \boldsymbol{\tau}^*_{n,i}(t)\dot{\boldsymbol{\tau}}_{n,i}(t), \quad \mathbf{T}_{n,0}(t) = \mathbf{W}_0 \tag{24}$$

$$\begin{aligned}\boldsymbol{\omega}(t) &= 2\mathbf{q}_n^*(t)\dot{\mathbf{q}}_n(t) = 2\mathbf{T}_{n,n} \\ &= 2(\mathbf{W}_i^*\mathbf{T}_{n,n-1}\mathbf{W}_i + \dot{\mathbf{W}}_i)\end{aligned} \tag{25}$$

In practice, the recursive approach is not only simpler to implement, but was shown by Sommer et al. [37] to be significantly faster to compute. Comparing our implementation using their benchmark also shows indistinguishable performance for automatic differentiation.

For a unit quaternion in scalar-vector form, $\mathbf{q} = (r, \mathbf{v})$, we define the scalar exponent and quaternion logarithm as:

$$\mathbf{q}^\alpha = \begin{pmatrix} \cos(\theta\alpha) \\ \sin(\theta\alpha)\frac{\mathbf{v}}{|\mathbf{v}|} \end{pmatrix}, \; \log(\mathbf{q}) = \begin{pmatrix} 0 \\ \theta\frac{\mathbf{v}}{|\mathbf{v}|} \end{pmatrix} \tag{26}$$

$$\theta = \begin{cases} \text{atan2}(|\mathbf{v}|, r) & r \geq 0 \\ \text{atan2}(-|\mathbf{v}|, -r) & \text{otherwise.} \end{cases} \tag{27}$$

These ensure $\theta \in [-\pi/2, \pi/2]$ i.e. that the shortest geodesic path is selected, regardless of the signs of $\mathbf{k}_i$. From this and (25) we see that regularizers for orientation splines like $\int |\dot{\boldsymbol{\omega}}(t)|^2 dt$ will lack analytical primitives in general.

3.3. The KKS Domain

The ambiguity resolved by (27) leads to an important but previously overlooked restriction on the domain of the KKS, in addition to those shared with the $\mathbb{R}$ spline. Using (25), $|\log(\mathbf{w}_i)| \leq \frac{\pi}{2}$, $|\mathbf{W}_i| = 1$ and the triangle inequality:

$$\begin{aligned}|\boldsymbol{\omega}(t)| \leq 2\sum_{i=1}^{n}|\dot{\mathbf{W}}_i(t)| = 2\sum_{i=1}^{n}|b_{n-1}(t-i)\log(\mathbf{w}_i)\mathbf{W}_i| \leq \\ 2\sum_{i=1}^{n} b_{n-1}(t-i)\frac{\pi}{2}|\mathbf{W}_i| = \pi\sum_{i=1}^{n} b_{n-1}(t-i) \leq \pi\end{aligned} \tag{28}$$

That is: $|\boldsymbol{\omega}(t)| \leq \pi$. More generally a spline with knot distance $\Delta t_i = 1$ has $|\boldsymbol{\omega}(t)| \leq \pi\Delta t_i^{-1}$. Thus for a spline to be able to express $\boldsymbol{\omega}(t)$, its knot distance must satisfy:

$$\Delta t_i \leq \pi|\boldsymbol{\omega}(t)|^{-1}, t \in \mathbb{R} \tag{29}$$

This novel constraint is necessary, but not sufficient, for the spline to be able to express a given dynamic. This can be used to regularize *e.g.* by assuming all $\boldsymbol{\omega}(t)$ above some value are noise. Another use case would be restricting the knot distance search given gyro measurements. In practice this means a sufficiently expressive knot distance can be found quickly. Note also that (29) implies a bound on the Riemann approximation error for many orientation regularizers.

4. Spline Endpoints and Extrapolation

In theory, the control points are often assumed to exist extending to infinity. In practice a finite range is used, and boundary effects must be considered for any regularizer of form $\int_{\mathbb{R}} f(s(t))dt$. In particular for (6) and (16), as they will extremely quickly move towards identity. The modified splines (7) and (17) alleviate, but do not resolve this issue, as their derivatives will still go to zero in just a few knots, which can also cause severe boundary effects. To address this, we propose a new class of pose trajectory spline which uses constant linear and angular velocity for extrapolation. Our new spline class has the finite control points $\mathbf{a}_i,\ i \in [B, E]$ and the full domain $\mathbb{R}$ while maintaining the continuity properties.

Using (9) we can expand $\ddot{\mathbf{s}}_n(t) = 0$ to:

$$\ddot{\mathbf{s}}_n(t) = \sum_i (\mathbf{a}_i - 2\mathbf{a}_{i-1} + \mathbf{a}_{i-2})b_{n-2}(t-i) = 0 \tag{30}$$

This constrains the control points as:

$$\mathbf{a}_i = 2\mathbf{a}_{i-1} - \mathbf{a}_{i-2} = \mathbf{a}_{i-1} + \Delta\mathbf{a}_{i-1} \tag{31}$$

which is a common recursive extrapolation strategy. Note however that (31) implies $\Delta\mathbf{a}_{i-1} = \mathbf{a}_i - \mathbf{a}_{i-1} = \Delta\mathbf{a}_i$, and thus $\Delta\mathbf{a}_i = \Delta\mathbf{a}_E$ for $i > E$. We can thus replace the recursion with:

$$\mathbf{a}_i = \mathbf{a}_E + (i-E)\Delta\mathbf{a}_E \text{ for } i > E\,. \tag{32}$$

Rather than using this expression, we replace the first and last basis functions with r_n, and d_n, and define the extrapolating cumulative spline as follows:

$$\begin{aligned}\mathbf{s}_n(t) =& \mathbf{a}_B + \Delta\mathbf{a}_{B+1}r_n(t-B) + \\ &+ \sum_{i=B+2}^{E-1} \Delta\mathbf{a}_i c_n(t-i) + \Delta\mathbf{a}_E d_n(t-E)\,.\end{aligned} \tag{33}$$

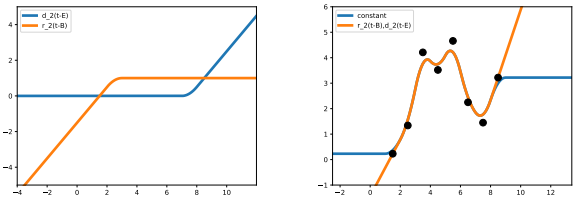

Figure 2. Left: Extrapolation functions (34) (blue) and (35) (orange) for $n = 2$ and $[B, E] = [0, 7]$. Right: Example extrapolation.

The extrapolation functions of degree n are:

$$d_n(t) = \sum_{i=0}^{\infty} c_n(t-i) \quad \text{and} \tag{34}$$

$$r_n(t) = 1 - d_n(n{+}1{-}t) = \sum_{i=-\infty}^{1} c_n(t-i)\,. \tag{35}$$

These two functions are plotted in figure 2. The forward expression can be rewritten for implementation as:

$$\begin{aligned} d_n(t) &= \sum_{i=0}^{\infty} c_n(t-i) \\ &= \begin{cases} \sum_{i=0}^{n-1} c_n(t-i) & \text{if } t < C \\ t - C + 1 & \text{otherwise.} \end{cases} \end{aligned} \tag{36}$$

Where $C = (n+1)/2$ is the center of gravity of the basis functions. Similarly, the backward function becomes:

$$\begin{aligned} r_n(t) &= \sum_{i=-\infty}^{1} c_n(t-i) - n \\ &= \begin{cases} \sum_{i=1-n}^{1} c_n(t-i) - n & \text{if } t > C \\ t - C & \text{otherwise.} \end{cases} \end{aligned} \tag{37}$$

We then replace linear space operations in (33) with corresponding quaternion operations as in Kim et al. [20] to obtain:

$$\mathbf{q}_n(t) = \mathbf{k}_B \mathbf{w}_{B+1}^{r_n(t-B)} \left[\prod_{i=B+2}^{E} \mathbf{w}_i^{c_n(t-i)} \right] \mathbf{w}_E^{d_n(t-E)} \tag{38}$$

This spline is parametrized by its control points $\{\mathbf{k}_i\}_B^E$, with the end-pairs $(\mathbf{w}_{B+1}, \mathbf{w}_E)$ defining extrapolating angular velocities. Similar to the $\mathbb{R}^N$ case, this smoothly converges to a constant angular velocity and maintains the continuity properties. The solution is convenient to use in practice as only the control point lookup and basis must be modified.

5. System Models and Regularizers

System models and regularizers are different sides of the same coin. Maximum likelihood fitting of a spline $\mathbf{s}(t, \Theta)$ implies minimizing a loss on the prediction errors of the observations $\mathbf{y}_l$ made at times t_l:

$$\mathcal{L}_{\text{obs}}(\Theta) = \sum_l \rho(\mathbf{y}_l - \mathbf{h}(\mathbf{s}(t_l, \Theta)))\,, \tag{39}$$

where ρ is the negative log of the error probability density and $\mathbf{h}$ is a measurement function. For e.g. the assumption of Gaussian errors, we have $\rho(\epsilon) = |\epsilon|^2$. We can regularize this cost by introducing an independent prior $\mathcal{L}_{\text{reg}}$, and a prior weight λ to get

$$\mathcal{L}(\Theta) = \mathcal{L}_{\text{obs}}(\Theta) + \lambda \mathcal{L}_{\text{reg}}(\Theta)\,. \tag{40}$$

Minimizing (40) results in a *maximum a posteriori* point estimate spline fit.

5.1. Derivative Based Regularizers

Derivative regularizers correspond to smooth functions e.g.

$$\mathcal{L}_{\text{der}}^m(\Theta) = \int_{-\infty}^{\infty} \left\| \frac{\partial^m \mathbf{s}(t, \Theta)}{\partial t^m} \right\|^2 dt\,. \tag{41}$$

This can be interpreted as an expectation that the chosen derivative is i.i.d. Gaussian with zero mean. E.g. $m = 2$ corresponds to a constant velocity model, but higher order derivative regularizers are in use as well, e.g. the $m = 4$ Snap [22]. For $\mathbb{R}^N$ splines such costs can be efficiently computed using their analytical primitives, as derived using our approach in (13).

5.2. Orientation Regularization

Similar to the $\mathbb{R}^N$ case, it is common to regularize orientation using derivatives of the angular velocity $\boldsymbol{\omega}(t)$ weighted by the inertial tensor $\mathbf{I}$ [14]:

$$\int_{\mathbb{R}} \frac{\partial^m \boldsymbol{\omega}(t)}{\partial t^m}^T \mathbf{I} \frac{\partial^m \boldsymbol{\omega}(t)}{\partial t^m} dt\,. \tag{42}$$

This regularizer lacks an analytical primitive in general.

5.3. Orientation P-spline

The P-spline regularizer $\mathcal{L}_{\text{P}}^m(\Theta) = \sum_i (\Delta^m \mathbf{a}_i)^2$ of Eilers and Marx [10] was originally derived as an approximation to (41), in part to simplify making a spline observable, and in part to avoid the "difficult" integral (13). This approach is popular due to its simplicity and is practical when little is known about the signal except that it is smooth.

The P-spline regularizer is limited to $\mathbb{R}^N$ splines. In order to address this, we here introduce the Orientation P-Spline regularizer $\mathcal{L}_{\text{PO}}(\Theta)$ which uses the unit quaternion Δ operator: $\Delta \mathbf{k}_i = \mathbf{k}_{i-1}^* \mathbf{k}_i$, $\Delta^m \mathbf{k}_i = \Delta^{m-1}(\Delta \mathbf{k}_i)$. This allows us to define the regularizer:

$$\mathcal{L}_{\text{PO}}^m(\Theta) = \sum_i |\Delta^m \mathbf{k}_i|_{\text{geo}}^2\,, \tag{43}$$

where we note that $|\mathbf{p}^*\mathbf{q}|_{\text{geo}} = 2\cos^{-1}(\mathbf{p}^T\mathbf{q})$. Thus $\mathcal{L}^1_{\text{PO}}(\Theta), \mathcal{L}^2_{\text{PO}}(\Theta)$ roughly correspond to a constant orientation and constant $\boldsymbol{\omega}(t)$ respectively. The first order in particular is useful for higher degree KKS splines, as it penalizes discontinuities, much like the corresponding $\mathbb{R}^N$ variant does.

5.4. Car Motion Models

There are two motion models commonly used for e.g. cars which we consider for the stereo visual odometry experiment section 7. In the vehicle coordinate system with z forwards, y down, x to the right and position p. A car moving forwards only leads to $\dot{p}_x = n(t)$, $\dot{p}_y = n(t)$ for iid Gaussian $n(t)$. Moving on a surface leads to the coordinated turn model $\ddot{p}_x = v_z(t)w_y(t) + n_0(t)$.

$$\mathcal{L}_{\text{fwd}}(\Theta) = \int_{\mathbb{R}} (\ddot{p}_x)^2 + (\ddot{p}_y)^2 dt \tag{44}$$

$$\mathcal{L}_{\text{ct}}(\Theta) = \int_{\mathbb{R}} (\ddot{p}_x - v_z(t)w_y(t))^2 \tag{45}$$

The latter cost (45) lacks analytic primitives in general, and many different approximations have been investigated, e.g. Roth [33] and Yuan [40]. The forward motion model has an analytic solution for specific pose definitions, but is nevertheless often approximated e.g. in SSBARv [39].

5.5. Integral Cost Approximation

In order to optimize over cost terms which are integrals, we must be able to either analytically evaluate them, or approximate them. A common approach is to approximate the function [14, 39], but we have noticed that for smooth functions such as splines, Riemann sums are an appealing alternative. In practice we should use few samples to minimize the computational cost. So how sparsely can we sample?

The key observation we make is that optimizing over samples rather than the integral can turn an overdetermined problem into a determined, or underdetermined one. This in turn allows the optimizer to overfit to the samples and thus perfectly satisfy the regularizer. This is the cause of the ringing artifacts shown in Figure 4. Avoiding this gives us a lower bound for the required minimum number of samples. For N knots we will need a minimum of $N + 1$ samples. This also implies that using one sample at each knot as in e.g. [28, 4], is insufficient. For a longer spline, 2 samples/knot is mostly sufficient, but for a short interval, and in the beginning, we should have at least $n + 2$ samples/knot to know that the state is overdetermined.

Next we consider the local frequency properties. When a function is sampled at a rate f_s, frequencies above $0.5f_s$ are aliased, and cause errors. The Fourier transform of the basis function is $\text{sinc}^n(f)$ which means that the amplitude of the aliased frequencies decay faster than $(0.5f_s)^{-n}$ where n is the degree (less the derivative) for e.g. (41). This also suggests a minimum sample rate above 2 per knot, and that the minimum approximation error drops faster for higher degree splines. Because the rotation spline is smooth and therefore can be locally described by a spline of the same degree with good precision, this is likely a good approximation for them as well.

We conclude that unless a careful constraint analysis has been performed, $n + 2$ evenly spaced samples per knot should be used.

6. Regularization Experiments

We evaluate how the sample rate affects estimate accuracy and compare the Orientation P-Spline and variable knot spacing. This experiment corresponds to a model of e.g. detected correspondences in RGB-D video. The ground truth trajectory uses a knot spacing of $\Delta t_i = 1$ and degree 4 and is generated by simulation. The trajectory smoothly rotates through, and between each of the planes while moving on a sphere, and is designed to expose every special case. Se the supplementary material for additional details. The regularizers are evaluated on the problem of fitting the pose spline trajectory $\hat{P}_{\boldsymbol{\Theta}}(t)$ consisting of KKS $q(t)$ and a $\mathbb{R}^3$ splines given 3D point correspondences which satisfy:

$$\hat{\mathbf{x}}_i = P(t)\mathbf{x} + \mathbf{n}_{i,t} \tag{46}$$

for ground truth trajectory $P(t)$ and the IID Gaussian noise $\mathbf{n}_{i,t} \sim \mathcal{N}(0, 1.0\mathbf{E})$. The 3D landmarks $\mathbf{x}$ are randomly drawn from the unit sphere at $5Hz$. We estimate a trajectory of the same degree and knot distance by minimizing:

$$\min_{\boldsymbol{\Theta}} \sum_{i,t} \alpha^{-1} \left\| \hat{\mathbf{x}}_i - \hat{P}_{\boldsymbol{\Theta}}(t)\mathbf{x} \right\|_2^2 + \beta^{-1}\mathcal{L}_{\text{reg}}(\boldsymbol{\Theta}) \tag{47}$$

where α, β are the variances of the measurements and of the regularizer on the ground truth. The estimated trajectory has the same degree and knot distance as the ground truth. The trajectory estimate is initialized using the ground truth, i.e. we are studying how far the noise causes the trajectory to diverge. This is to avoid any issue caused by local minima. As metric we use $\int \theta^2(t)dt$ where θ is the angle between ground truth and estimated orientations. The integral is evaluated using dense sampling (10^5 per knot).

Figure 3 shows the relative accuracy of the constant angular velocity model, often called *energy* (i.e. (41) with $n = 2$) when varying the number of regularizer samples that are used per control point.

The figure shows that the energy regularizer provides excellent performance even at low sampling rates. Qualitatively the reconstruction is nearly indistinguishable from the ground truth for $f_s \geq 2$ whereas the signal is extremely noisy without regularization. Using one sample per knot, the system performs reasonably well, but visibly oscillates/overfits at the beginning as shown in Figure 4 and predicted by the constraint analysis.

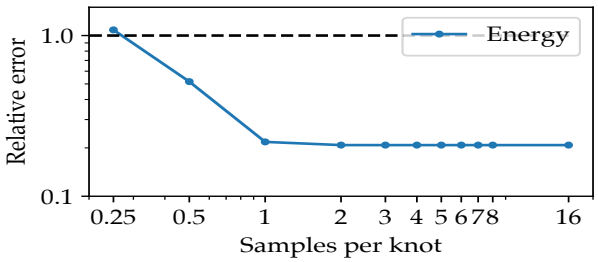

Figure 3. Performance of the regularizers depending on the number of samples per spline knot that are used for the approximating integral. The error is reported relative to using no regularization. Knot spacing $\Delta t_i = 1$.

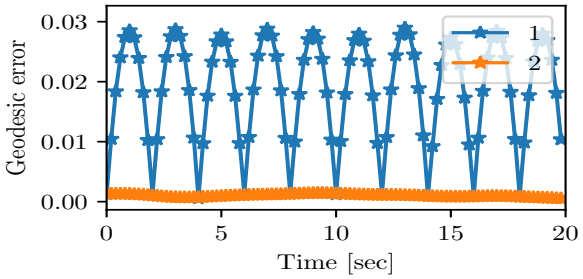

Figure 4. Sampling the regularizer using single sample per knot allows the optimizer to overfit, exactly satisfying the samples and causing oscillation if the regularizer weight is high enough.

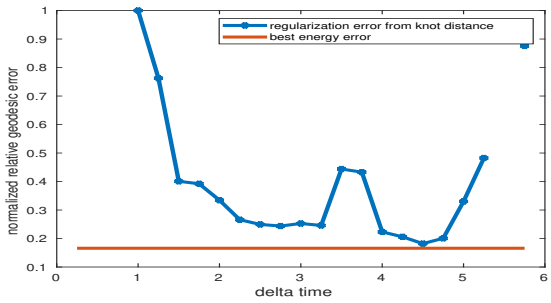

Figure 5. Regularization using knot spacing as an implicit regularizer. The error is reported relative to using no regularization and knot spacing $\Delta t_i = 1$. The difference to the energy regularization is approximately 30%. The problem becomes unobservable for knot spacing below 0.2 and errors above 1 are excluded.

We also compare to implicit regularization by using varied knot spacing in figure 5. We conclude that if the optimal knot spacing can be found, this approach works reasonably well, but it is difficult to predict what that spacing is in advance based on e.g. dynamic models.

For P-spline regularization (43) we try using both first and second order, and vary the regularizer weight. In both cases the optimal weight is predicted by the ground truth

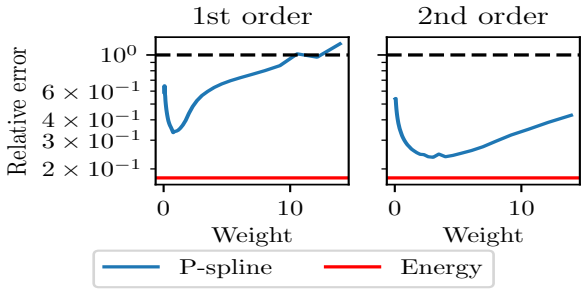

Figure 6. Performance of P-spline regularizers for different weights. Left: $m = 1$, right: $m = 2$. The error is reported relative to using no regularization and knot spacing $\Delta t_i = 1$.

variance weighting. Figure 6 shows that second order generalized P-splines are better than first order, but both are inferior to using the sampled energy model.

We conclude that the energy penalty with a moderate number of samples per knot performs well. The improvement plateaus after 2 samples per knot. Adding this regularizer significantly improves convergence speed, reducing total computational time by 33%. Overall, the generalized P-splines are faster but also perform worse.

7. Visual Odometry

This experiment demonstrates that continuous pose trajectories are applicable to real-time odometry using the KITTI odometry dataset [16]. Specifically we examine how varying knot distance and spline degree affects the computation time and reconstruction accuracy.

The visual odometry system we use is intentionally simple. We use a KLT [32] tracker and a SGM [17] stereo, which at each step tracks 3d features into the next image. After minimization we discard outliers based on a reprojection error with threshold $3\,pixels$ by eliminating the track's last measurement. We initialize new features using ANMS [15] and prune old features using the approach of [31] keeping the number of tracks $\in [100, 300]$. The system is initialized using PnP for the first five frames, after which the regularizers allow the spline to predict the next frame pose. In each frame the system optimizes the reprojection errors, the sampled regularization using n+2 samples, and the 3d point positions, giving the cost:

$$\mathcal{L}(\Theta) = \sum_{k,f} \|y_{t_i,f} - \wp(P_{cv} P_{wv}^{-1}(t_i) x_f)\|^2 \tag{48}$$

$$+ \alpha^{-1} \mathcal{L}_{\text{fwd}}(\Theta) + \beta^{-1} \mathcal{L}_{\text{ct}}(\Theta) \tag{49}$$

where $y_{t_i,f}$ are observations of feature f at time t_i and $\wp$ is the stereo projection. P_{cv} is the transform between the camera and the rear axle, and $P_{wv}(t)$ is the pose trajectory. The weights α, β are found as the inverse variances of the regularizers applied to the ground truth trajectory of KITTI

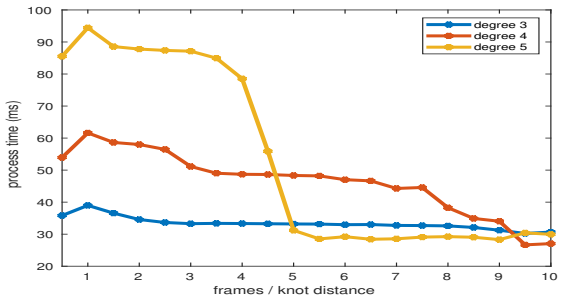

Figure 7. The processing time per frame related to the knot distance, for $n = 3, 4, 5$ on knot distances from 0.05 to 1. Evaluated on the first training sequence.

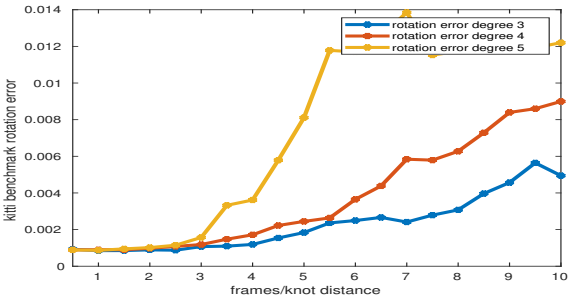

Figure 8. The KITTI rotation error metric evaluated on the first training sequence, for $n = 3, 4, 5$ on knot distances from 0.05 to 1. The current top performer SOFT2[7] achieves an error of 0.0009 overall on test data.

sequence two. The entire history of each track present in any of the five most recent frames are included in the cost. All knots preceding those affected by the five most recent frames are set constant, as are all 3d points which are not present in the current image. We use up to five iterations per frame. To our knowledge, we are the first to use a windowed optimization strategy for real-time VO on real data.

We evaluate the system on the first training sequence as it lets us perform the evaluation repeatedly and because it is simple enough to use a simple tracker. Results are shown in figures 7 and 8. This demonstrates that real-time online stereo odometry using pose trajectory splines is feasible, with the degree three alternative running at 30fps while achieving reasonable results. The experiment also shows that knot distances shorter than one are possible given that a regularizer is used. A video showing sample results is available in the supplementary material.

8. Visual-Inertial Calibration

Visual-inertial calibration is commonly used as a validation experiment for pose trajectory spline implementations, e.g. Furgale et al. [14], Patron-Perez et al. [28] and Sommer et al. [37]. We similarly validate our system by calibrating the *calib-cam1* sequence from the dataset by Shubert et al. [35]. The sequence has 10 336 IMU measurements and 1038 stereo pairs spanning 52.8 seconds, but it lacks ground truth. Therefore we compare to the result of Sommer et al. [37] and use the same $P_{\text{cam,imu}}$ initialization and weights $\gamma_{\text{acc}}, \gamma_{\text{gyro}}$ as them. We initialize the spline using per frame PnP-Ransac [30, 11] followed by bundle adjustment.

Since poses from calibration patterns are highly accurate, we wish to be able to go through each, which implies a minimum knot distance of $0.05s$. We verify that this is likely sufficient to represent the observed $\boldsymbol{\omega}(t)$ as using (29) implies a minimum knot distance of $0.7s$. The lowest derivative of interest i.e. acceleration should have one or preferably two continuous derivatives. Therefore degree 3 or 4 is suitable and we use four. We find the transform P_{ci} between camera and IMU by minimizing the cost:

$$\mathcal{L}(\Theta) = \sum_{t_f,i} \|y_{t_f,i} - \wp(P_{cw}(t_f)x_i)\|^2 + \tag{50}$$

$$\gamma_{\text{acc}} \sum_{t_a} \|\ddot{p}_{\text{body}}(t_a) - \ddot{p}[t_a] + g_{\text{body}} + b_{\text{acc}}\|^2 + \tag{51}$$

$$\gamma_{\text{gyro}} \sum_{t_g} \|\omega_{\text{body}}(t_g) - \omega_{\text{body}}[t_g] + b_{\text{gyro}}\|^2 \tag{52}$$

Where $\wp$ is the projection operator, $y_{t_f,i}$ visual measurements, and $P_{cw}(t) = P_{ci}P_{wi}^{-1}(t)$ the offset pose trajectory spline. The $\omega_{\text{body}}(t_i)$, and $\ddot{p}(t_i)$ are gyro and accelerometer measurements with constant IMU bias. We optimize over P_{ci} and the pose trajectory. We do not add regularization, as this would conflict with the sought estimate.

Using this procedure we recover the IMU-camera transformation with a geodesic difference of 10^{-5} and 10^{-4} meters, relative to the calibration of Sommer (which is not a ground truth).

9. Concluding Remarks

This paper has investigated pose trajectory regularization by sampling explicit non-linear priors. This is aided by a novel spline construct with constant linear and angular velocity extrapolation. We have found a lower bound for the sampling rate, which is notably above the common choice of a single sample per knot. We have compared several different types of pose trajectory regularization, including derivative penalties, implicit (using knot spacing), and a novel orientation P-spline penalty. We have also derived an upper bound on the knot distance required to represent a given $\boldsymbol{\omega}(t)$. The code for our system is available on GitHub [29]. The system has been validated on the standard task of visual-inertial calibration, and applied to stereo visual odometry where we demonstrate real-time performance.

Acknowledgements: This work was supported by Vinnova through the Visual Sweden network, Dnr 2019-02261, and by Singulareye.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. 1

[2] Sean Anderson and Timothy D. Barfoot. Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on se(3). In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 157–164, 2015. 2

[3] Sean Anderson, Frank Dellaert, and Timothy D. Barfoot. A hierarchical wavelet decomposition for continuous-time SLAM. In *IEEE International Conference on Robotics and Automation (ICRA14)*, pages 373–380, Hong Kong, China, May 2014. 2

[4] Sean Anderson, Kirk MacTavish, and Timothy D. Barfoot. Relative continuous-time SLAM. *International Journal of Robotics Research*, 34(12):14531479, 2015. 1, 2, 6

[5] T. Barfoot, C. Tong, and S. Särkkä. Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Robotics: Science and Systems*, 2014. 2

[6] Charles Bibby and Ian D. Reid. A hybrid SLAM representation for dynamic marine environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 257–264. IEEE, 2010. 2

[7] Igor Cvii and Ivan Petrovi. Stereo odometry based on careful feature selection and tracking. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2015. 8

[8] Carl de Boor. *A Practical Guide to Splines*. Springer Verlag (New York), 1978. 2

[9] Jing Dong, Mustafa Mukadam, Byron Boots, and Frank 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)*, pages 6497–6504, 2018. 2

[10] Paul H. C. Eilers and Brian D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996. 5

[11] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the Association for Computing Machinery*, 24(6):381–395, 1981. 8

[12] Paul Furgale, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch estimation using temporal basis functions. In *IEEE International Conference on Robotics and Automation (ICRA12)*, 2012. 1

[13] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE International Conference on Intelligent Robots and Systems (IROS13)*, Tokyo, Japan, November 2013. 2

[14] Paul Furgale, Chi Hay Tong, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch trajectory estimation using temporal basis functions. *International Journal of Robotics Research*, 34(14):1688–1710, 2015. 1, 2, 3, 5, 6, 8

[15] S. Gauglitz, L. Foschini, M. Turk, and T. Hollerer. Efficiently selecting spatially distributed keypoints for visual tracking. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1869–1872, Sept 2011. 7

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[17] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 7

[18] Kun Huang, Yifu Wang, and Laurent Kneip. B-splines for purely vision-based localization and mapping on non-holonomic ground vehicles. In *IEEE International Conference on Robotics and Automation (ICRA21)*, 2021. 2

[19] Christian Kerl, Jörg Stückler, and Daniel Cremers. Dense continuous-time tracking and mapping with rolling shutter cameras. In *IEEE International Conference on Computer Vision (ICCV15)*, 2015. 2

[20] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *SIGGRAPH'95*, pages 369–376, 1995. 1, 2, 3, 4, 5

[21] Steven Lovegrove, Alonso Patron-Perez, and Gabe Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *British Machine Vision Conference (BMVC)*. BMVA, September 2013. 1, 2

[22] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011. 5

[23] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018. 1, 2

[24] Luc Oth, Paul Furgale, Laurent Kneip, and Roland Siegwart. Rolling shutter camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR13)*, pages 1360–1367, Portland, Oregon, June 2013. 1, 2

[25] Hannes Ovrén and Per-Erik Forssén. Spline error weighting for robust visual-inertial fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, June 2018. Computer Vision Foundation. 1, 2

[26] Hannes Ovrén and Per-Erik Forssén. Trajectory representation and landmark projection for continuous-time structure from motion. *International Journal of Robotics Research*, 38(6):686–701, May 2019. 1, 2

[27] Chanoh Park, Peyman Moghadam, Soohwan Kim, Alberto Elfes, Clinton Fookes, and Sridha Sridharan. Elastic LiDAR fusion: Dense map-centric continuous-time SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 2

[28] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal on Computer Vision*, 113(3):208–219, 2015. 1, 2, 6, 8

[29] Mikael Persson. Explicitly regularized pose splines. https://github.com/midjji/pose-splines, 2021. 1, 8

[30] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (P3P) solver. In *The European Conference on Computer Vision (ECCV)*, September 2018. 8

[31] M Persson, T Piccini, R Mester, and M Felsberg. Robust stereo visual odometry from monocular techniques. In *IEEE Intelligent Vehicles Symposium*, 2015. 1, 7

[32] Clemens Rabe. *Detection of Moving Objects by Spatio-Temporal Motion Analysis: Real-time Motion Estimation for Driver Assistance Systems*. Südwestdeutscher Verlag für Hochschulschriften, 2012. ISBN 978-3-8381-3219-8. 7

[33] Michael Roth, Gustaf Hendeby, and Fredrik Gustafsson. Ekf/ukf maneuvering target tracking using coordinated turn models with polar/cartesian velocity. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8, 2014. 6

[34] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[35] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The TUM VI benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018. 8

[36] Ken Shoemake. Animating rotation with quaternion curves. In *Int. Conf. on CGIT*, pages 245–254, 1985. 1

[37] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative B-splines on Lie groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR20)*, June 2020. 1, 2, 4, 8

[38] Michael Unser. Splines – a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, 1999. 2, 3

[39] Xin Wang, Fei Xue, Zike Yan, Wei Dong, Qiuyuan Wang, and Hongbin Zha. Continuous-time stereo visual odometry based on dynamics model. In *14th Asian Conference on Computer Vision (ACCV18)*, volume 11366 of *Lecture Notes in Computer Science*, pages 388–403. Springer, 2018. 2, 6

[40] Xianghui Yuan, Chongzhao Han, Zhansheng Duan, and Ming Lei. Comparison and choice of models in tracking target with coordinated turn motion. In *2005 7th International Conference on Information Fusion*, volume 2, pages 6 pp.–, 2005. 6