

Modified Rodrigues Parameters: An Efficient Representation of Orientation in 3D Vision and Graphics

George Terzakis¹  · Manolis Lourakis² · Djamel Ait-Boudaoud³

Received: 6 May 2017 / Accepted: 17 September 2017 / Published online: 12 October 2017
© The Author(s) 2017. This article is an open access publication

Abstract Modified Rodrigues parameters (MRPs) are triplets in \mathbb{R}^3 bijectively and rationally mapped to quaternions through stereographic projection. We present here a compelling case for MRPs as a minimal degree-of-freedom parameterization of orientation through novel solutions to prominent problems in the fields of 3D vision and computer graphics. In our primary contribution, we show that the derivatives of a unit quaternion in terms of its MRPs are simple polynomial expressions of its scalar and vector part. Furthermore, we show that updates to unit quaternions from perturbations in parameter space can be computed without explicitly invoking the parameters in the computations. Based on the former, we introduce a novel approach for designing orientation splines by configuring their back-projections in 3D space. Finally, in the general topic of nonlinear optimization for geometric vision, we run performance analyses and provide comparisons on the convergence behavior of MRP parameterizations on the tasks of absolute orientation, exterior orientation and large-scale bundle adjustment of public datasets.

Keywords Orientation parameterization · Quaternions · Stereographic projection · Modified Rodrigues parameters · Orientation interpolation · Quaternion interpolation · Structure from motion · Nonlinear optimization

1 Introduction

Orientation or attitude is a prominent facet of problems pertaining to disciplines such as computer graphics, computer vision, photogrammetry, robotics and augmented reality. A typical example of such a problem in computer animation is the interpolation of orientation during a process known as key-framing, the main objective of which is to achieve esthetically pleasing results in the representation of the motion of a rigid object not only in terms of translation, but also in terms of the changes in its orientation [66].

In an inverse fashion, 3D computer vision deals with 3D reconstruction, also often referred to as structure from motion estimation (SfM). This consists in using sets of images depicting an unknown scene and captured from unknown locations, in order to automatically extract a 3D geometric representation of the imaged scene plus the camera intrinsic parameters and their poses, i.e., positions and orientations [26,45]. Several solutions to the SfM problem involve the estimation of the sought parameters by iteratively minimizing the total geometric error pertaining to overdetermined sets of image measurements. Considering that a 3D rotation matrix has nine elements but only three degrees of freedom (DoF), suitable (and preferably minimal) parameterizations of rotation are thus necessary in order to intrinsically incorporate orthonormality constraints on rotations during the optimization.

The modified Rodrigues parameters (MRPs) constitute a minimal rotation parameterization with attractive properties.

✉ George Terzakis
george.terzakis@port.ac.uk

Manolis Lourakis
lourakis@ics.forth.gr

Djamel Ait-Boudaoud
dab@port.ac.uk

¹ Faculty of Technology, University of Portsmouth, LG204, Lion Terrace, Portsmouth PO1 3HF, UK

² Institute of Computer Science, Foundation for Research and Technology - Hellas, N. Plastira 100, 70013 Heraklion, Greece

³ Faculty of Technology, University of Portsmouth, Portland Building, Portland Terrace, Portsmouth PO1 3AH, UK

Despite being well-established in the field of aerospace engineering, MRPs are, to the best of our knowledge, unheard-of in the computer graphics, vision and robotics communities. MRPs are essentially the stereographic coordinates of quaternions and, as such, they are mapped rationally and bijectively to the quaternion sphere. This paper studies the representation of orientation via MRPs. An important finding is that the Jacobian of a quaternion is a polynomial function of its scalar and vector parts, thereby yielding simple expressions in rotation derivatives. Furthermore, it is shown that quaternions can be updated from a given perturbation in parameter space without explicitly using the MRPs. These two findings are very important for iterative optimization, because they allow both Jacobian computation and orientation updates to be carried out using exclusively quaternion components in simple additions and multiplications. As a consequence, iterative optimization completes with fewer calculations in less time. The paper also demonstrates the applicability of MRPs in problems related to orientation interpolation and pose estimation and provides experimental evidence that their use leads to new solutions or the simplification of existing ones and, in most cases, the improvement of performance.

The rest of the paper is structured as follows. A brief overview of orientation representations with respect to various applications is given in Sect. 2, followed by descriptions of common problems involving parameterized orientation and respective solutions in Sect. 3. A derivation of the MRPs based on stereographic projection is provided in Sect. 4, whereas Sect. 5 discusses special properties of MRPs as vectors parallel to the rotation axis; such properties include the relationship with axis-angle and Gibbs vectors as well as the Cayley transform from MRPs to rotation matrices. Section 6 focuses on the differentiation properties of the MRP parameterization. In Sect. 7, the problem of smooth interpolation on the quaternion sphere is examined and a general method for spline based fitting is presented. Experimental results comparing the performance of MRPs against different parameterizations of rotations are given in Sects. 8 and 9 summarizes the contributions of the paper.

2 Rotation Representations

The literature provides many representations of rotation using 3×3 matrices and vectors of three or four components. Representing spatial rotations is challenging due to their non commutativity and the fact that their topology does not permit a smooth embedding in Euclidean 3D space. Besides, different practical uses of rotations have different requirements. This section briefly presents the representations most commonly employed in the fields of graphics,

robotics and vision, and discusses their strengths and weaknesses.

2.1 Rotation Matrices

Rotation matrices are 3×3 orthonormal matrices that arguably constitute the most intuitive representation of orientation. The reason for this is that ordinary linear algebra can be employed to express common operations involving rotations. For example, a point can be rotated using standard matrix-vector multiplication, two rotations can be composed via matrix multiplication, whereas a rotation can be inverted via matrix transposition. A rotation matrix consists of nine elements but has only three DoFs due to the six independent constraints imposed by orthonormality.

Being quadratic, these constraints are cumbersome to impose, typically in the context of a Lagrangian function. This, however, does not render the representation entirely unattractive. For instance, Carlone et al. [7], Olsson and Eriksson [43] as well as Briales and Jimenez [6] make explicit use of matrix orthonormality constraints to formulate the Lagrangian of the camera pose registration problem. The advantage of this approach is that it provides measures for the optimality of solutions of relaxations by monitoring the duality gap in the original problem.

In overall, rotation matrices are typically used to represent rotations when transformations of objects such as points and lines are involved but not very often used in other operations such as interpolation and estimation. For future reference, the set of all rotation matrices is the 3D rotation group, denoted $SO(3)$.

2.2 Euler Angles

Euler angles define a rotation in terms of three consecutive elemental rotations around the orthogonal axes of a Cartesian coordinate system. There exist twelve possible sets of Euler angles, depending on the chosen, non commutative order of rotation axes. Even for a particular axes sequence, Euler angles are not unique since supplementary and/or negative angles can yield the same overall rotation [58]. Although an easily conceptualized and minimal DoF encoding, there exist several arguments as to why Euler angles are a parameterization scheme unsuitable for most applications [16, 28, 51].

Notwithstanding their ambiguity, the primary drawback of Euler angles is that they suffer from singularities near which infinitesimal changes in orientation can cause large jumps in the values of their elemental constituent rotations [51, 58]. When represented with Euler angles, every orientation is at most 90 degrees away from a singularity. Such a singularity, known as gimbal lock from its physical manifestation in gyroscopes, occurs when two of the

three rotation axes coincide and results in the loss of one degree of freedom, i.e., one rotation having no effect [44]. Since gimbal lock is a discontinuity in the Euler angle representation, it might have undesirable side-effects such as ill-conditioning or instabilities in applications involving rotation operations like iterative optimization, filtering, averaging or interpolation. Thus, the use of Euler angles in describing large and especially arbitrary rotations is limited. Furthermore, the kinematic differential equations of Euler angles are fairly nonlinear, involving computationally expensive trigonometric functions [51]. On the other hand, being more understandable to humans, Euler angles are commonly used in user interfaces for 3D rotations in graphics and CAD software. Another favorable application concerns the use of Euler angles with a linear Kalman filter for position and orientation tracking, as they maintain a linear process model.

2.3 Axis-Angle

Every rotation can be represented as a revolution by an angle θ around an axis parallel to a unit 3-vector u . The vector $\omega = \theta u$ is the angle-axis representation of a rotation. This representation is not unique, since an equivalent representation for the same rotation is $-(2\pi - \theta)u$. The matrix representation in $\mathcal{SO}(3)$ of an axis-angle rotation ω is given by the infinite series

$$\exp([\omega]_{\times}) = \mathbf{I} + \sum_{n=1}^{\infty} \frac{1}{n!} [\omega]_{\times}^n, \quad (1)$$

where $[\omega]_{\times}$ is the cross-product skew symmetric matrix associated with $\omega \stackrel{\text{def}}{=} [\omega_1 \ \omega_2 \ \omega_3]^T$:

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Equation 1 is referred to as the exponential map and can be evaluated with Rodrigues' formula for $\theta = \|\omega\|$ [48]:

$$R(\omega) = \exp([\omega]_{\times}) = \mathbf{I} + \frac{\sin(\theta)}{\theta} [\omega]_{\times} + \frac{1 - \cos(\theta)}{\theta^2} [\omega]_{\times}^2 \quad (2)$$

The axis-angle representation is a many-to-one mapping and has singularities at $\theta = 2n\pi$, $n \geq 1$. Furthermore, although the formula converges to the identity matrix for very small values of θ , in practice it presents numerical issues which call for approximating $\sin(\theta)$ and $\cos(\theta)$ with their Taylor series expansions near the origin and using them to simplify the two fractions in Eq. 2. On the other hand, it is surjective,

i.e., every rotation has a representation as the exponential of a skew symmetric matrix. Since it is minimal and does not require any additional constraints, the axis-angle representation is very often employed in vision and robotics problems. The terms axis-angle and exponential map are used interchangeably.

Although the exponential map is periodic, a *rotation matrix logarithm*¹ from rotation matrices to axis-angle vectors can be defined for $\theta \in (-\pi, \pi)$:

$$\log R = \frac{\theta}{2 \sin \theta} (R - R^T) \quad (3)$$

$$\theta = \arccos\left(\frac{\text{Tr}(R) - 1}{2}\right), \quad (4)$$

where $\text{Tr}()$ denotes a square matrix's trace, i.e., the sum of its diagonal elements. Evidently, the rotation matrix logarithm converges to the zero vector at the identity, but in practice it is necessary to resort to approximations in order to avoid the effects of very small numbers in the denominator.

In the specific case where Eq. 4 yields $\theta = \pi$, the axis-angle vector cannot be recovered with Eq. 3. Instead, we initially obtain the absolute values of the components of ω as follows:

$$|\omega_i| = \sqrt{\frac{r_{ii} + 1}{2}}, \quad (5)$$

where $i \in \{1, 2, 3\}$ and r_{ij} is the element of R in the i th row and j th column. Since ω is sign-ambiguous, we may choose the component which has the largest absolute value to be positive. The remaining two components can be recovered from the off-diagonal elements of R as follows:

$$\omega_j = \frac{r_{kj}}{\omega_k}, \quad (6)$$

where $k = \arg \max_i \{|\omega_i|\}$ and $j \in \{1, 2, 3\} - \{k\}$.

2.4 Unit Quaternions

One of the most popular ways to unambiguously represent orientation in 3D is with the Euler–Rodrigues parameters, as a location on the unit sphere in 4D. The modern formalism for Euler–Rodrigues parameters are unit quaternions, which form a multiplicative group that fully describes 3D rotations. Unit quaternions constitute a redundant parameterization which does not suffer from gimbal lock. However, their numerical estimation in practice is complicated by the need to incorporate a unit-norm constraint; more details on this are provided in Sect. 3.2.

¹ The formula in Eq. 3 is readily obtained by taking the difference $R - R^T$ using Rodrigues' formula. Similarly, the angle in Eq. 4 is obtained by taking the trace of $R + R^T$.

For consistency of notation throughout the rest of the paper, the field of quaternions will henceforth be denoted with \mathbb{H} , and an arbitrary quaternion $q \in \mathbb{H}$ will be written in the form

$$q = \rho + v^T \varphi, \tag{7}$$

where $\rho \in \mathbb{R}$ is the *scalar part*, $v \in \mathbb{R}^3$ is the *vector part* and $\varphi = [i \ j \ k]^T$ is the vector of the 3 *imaginary units* (also referred to as *fundamental quaternion units*). The reader is referred to [16,29,44,61,64] for more detailed introductions on quaternions and their properties.

A quaternion $q \in \mathbb{H}$ such that $\|q\| = 1$, is called a *unit quaternion*. Unit quaternions lie on the unit sphere in 4D (also known as the *quaternion sphere* in kinematics or 3-sphere \mathbb{S}^3 in topology) and form a group under multiplication which precisely describes the group of rotations. In particular, provided a quaternion $q = \rho + v^T \varphi$ such that $\rho^2 + v^T v = 1$, it can be shown that it corresponds to the following rotation matrix [41]:

$$R = (\rho^2 - v^T v) I_3 + 2vv^T + 2\rho[v]_{\times} \tag{8}$$

It is worth noting that the formula in Eq. 8 implies that the elements of the rotation matrix are polynomial expressions of the unit quaternion components. It is also evident from Eq. 8 that the same rotation matrix corresponds to quaternions q and $-q$. Thus, when treated as elements of the special 3×3 orthogonal group, antipodal unit quaternions represent the same rotation.

In direct analogy to the exponential map from the space of skew-symmetric matrices to the group of rotation matrices, there exists an exponential map from the space of axis-angle vectors to unit quaternions:

$$\exp\left(\frac{1}{2}\omega^T \varphi\right) = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \left(\frac{\omega}{\theta}\right)^T, \tag{9}$$

where $\omega \in \mathbb{R}^3$ is the axis-angle vector or the rotation associated with q and $\theta = \|\omega\|$. Note that the right-hand side of Eq. 9 can be obtained with the Taylor expansion of $\exp\left(\frac{1}{2}\omega^T \varphi\right)$, provided the observation $(\omega^T \varphi)^2 = -\theta^2$.

As in the case of rotation matrices, unit quaternions can be mapped to the corresponding axis-angle vectors via a logarithmic function.²

$$\log q = \frac{\theta}{2 \sin \frac{\theta}{2}} (q - \bar{q}) \tag{10}$$

$$\theta = 2 \arccos \left(\frac{q + \bar{q}}{2} \right), \tag{11}$$

where \bar{q} denotes the conjugate of q , i.e., $\bar{q} = \rho - v^T \varphi$.

² Formulas are derived from Eq. 9 by considering $q - \bar{q}$ and $q + \bar{q}$ respectively.

3 Common Problems and Standard Solutions

This section briefly describes prevalent problems involving parameterized orientation along with the most common solutions employed and their typical shortcomings. Although applications may vary, these problems essentially fall under two major categories, namely *interpolation of orientation* and *estimation of rotation parameters*.

3.1 Interpolation of Orientation

The problem of interpolating rotational motion from a sequence of *key orientations* often arises in computer animation, computer-aided design and robot kinematics applications [11,27,30,49,53,59]. Since quaternions conveniently possess the properties of a metric space (i.e., \mathbb{R}^4), it is very common to perform this task on the unit sphere in 4D where properties such as smoothness, length and curvature can be measured and manipulated with standard calculus.

Suppose that a sequence of key orientations is given in the form of unit quaternions $q_0, q_1, \dots, q_n, \dots$ and the goal is to interpolate the sequence with a smooth spherical curve. Possibly the most popular tool for elementary interpolation on a great arc between two successive quaternions is Shoemake’s classic formula for *spherical linear interpolation* (abbreviated as SLERP) [56]:

$$\text{SLERP}(q_n, q_{n+1}; u) = \frac{\sin(1-u)\Phi}{\sin \Phi} q_n + \frac{\sin u\Phi}{\sin \Phi} q_{n+1}, \tag{12}$$

where $u \in [0, 1]$ is the interpolation parameter, $\Phi = \arccos(q_n \cdot q_{n+1})$ is the angle between q_n, q_{n+1} and \cdot denotes the dot product between quaternions as vectors in \mathbb{R}^4 .

Interpolation of more than two key orientations is a far more challenging task, primarily because the constituent segments of the curve have to be pieced smoothly at the data points. A popular solution is Shoemake’s spherical quadrangle interpolation (SQUAD) [57]. In a nutshell, SQUAD is the spherical analog of parabolic blending between quaternions q_n and q_{n+1} :

$$\text{SQUAD}(q_n, q_{n+1}; u) = \text{SLERP}\left(\text{SLERP}(q_n, q_{n+1}; u), \text{SLERP}(\alpha_n, \alpha_{n+1}; u(1-u))\right), \tag{13}$$

where $u \in [0, 1]$ is the interpolation parameter and α_n, α_{n+1} are auxiliary points chosen specifically to impose smoothness at the key points that can be computed with the following formula:

$$\alpha_i = q_i \exp\left(-\frac{\log(\bar{q}_i q_{i-1}) + \log(\bar{q}_i q_{i+1})}{4}\right) \tag{14}$$

Shortly after Shoemake's contribution, Duff followed with [20], introducing a B-spline spherical curve interpolating the data similarly to planar B-splines. More recently, a construction scheme in \mathbb{R}^3 for smooth quaternion curves was proposed in [32]. The main shortcoming associated with the aforementioned methods is the lack of a general strategy to enforce arc length and curvature minimization constraints, owing to the complexity of the corresponding expressions for the spherical polynomial derivatives. In their thorough report on Shoemake's work, Dam et al. [13] show that finding a generic, curvature minimizing smooth exponential curve on the sphere is highly impractical, primarily due to the complicated derivatives of the curve. Another ramification of the generally intractable differentiation is that in most cases, speed adjustment is performed purely numerically (cf. the *chord-length approximation* method [67]).

Following Shoemake's work, several solutions for orientation interpolation have been introduced, taking matters from a different perspective. For instance, Johnstone and Williams [31] introduced a rational function mapping 4D Bézier curves onto the quaternion sphere. Although they were unaware at the time, this mapping is the generalized form of stereographic projection [17], which, without any precautions will cause distortions on the sphere. Other geometric methods were proposed in [3,46,47]. With the exception of the method by Roberts et al. [47], these approaches focus on minimizing functionals defined on characteristics of the curve such as tangential velocity or centrifugal acceleration primarily by making approximations to the actual expressions. The common drawback of these approaches is that they are relatively complex to implement and not so flexible to configure under different circumstances.

An interesting alternative to the mainstream is the work of Boumal [5], which optimizes a cost function directly over rotation matrices. Since it is difficult to fit a parametric function on matrices with orthonormal constraints, Boumal defines a cost function over a sequence of rotations with penalty terms on chordal distance³ from the key-rotations as well as on first- and second-order finite differences in $SO(n)$ to impose smoothness. To iteratively optimize the cost function on the rotation manifold, analytical expressions for the Riemannian derivatives of the penalty terms are obtained as orthogonal projections of Euclidean matrix derivatives onto the tangent space of the current rotation estimates [1]. The method can be adapted either for interpolation or regression and applies to problems involving orthogonal matrices of arbitrary dimensionality.

³ The Frobenius norm of the difference of rotation matrices.

3.2 Estimation of Orientation and Rotation Matrix Differentiation

At the very core of several key problems in computer graphics, vision and robotics lies the problem of estimating orientation. The typical formulation of orientation estimation problems involves a cost function which is a sum of positive (by means of a suitable metric) error terms, in which the unknown rotation matrices act on vectors measured in different coordinate frames. The usual method of minimizing such a cost function is by setting its derivatives equal to zero and solving the resulting equations. The Jacobian of the rotation matrix is therefore crucial to the estimation.

It becomes evident from the exponential map expression in Eq. 1 that differentiation of the rotation matrix R with respect to the axis-angle vector ω is not trivial and the associated Jacobian contains complicated transcendental expressions. Most importantly, the derivative of the exponential map presents a "malignant" singularity at the origin, owed to the presence of an angle in the denominator. A complete list of analytic expressions for these derivatives can be found in a report by Diebel [16].

Recently, Gallego and Yezzi [23] have discovered a reasonably compact expression for the Jacobian of the rotation matrix:

$$\frac{\partial R}{\partial \omega_i} = \frac{\omega_i [\omega]_{\times} + [\omega \times (I_3 - R) e_i]_{\times}}{\theta^2} R, \quad (15)$$

where $i \in \{1, 2, 3\}$ indexes the components of ω and e_i is the i th canonical basis vector of \mathbb{R}^3 . It should be stressed that despite the denominator θ^2 in Eq. 15, the derivatives of the rotation matrix are continuous at the origin, $\omega = [0 \ 0 \ 0]^T$ and are equal to the cross-product skew symmetric matrices associated with the canonical vectors e_i (also known as the *infinitesimal generators* of the Lie algebra $so(3)$ and denoted G_i) [54]:

$$\left. \frac{\partial R}{\partial \omega_i} \right|_{\omega_i=0} = [e_i]_{\times} \stackrel{\text{def}}{=} G_i \quad (16)$$

Gallego and Yezzi's formula in Eq. 15 is a significant improvement, yet it still is not simple enough, let alone it entails the evaluation of a few trigonometric expressions. Furthermore, the singularity at the origin must still be accounted for with the aid of Taylor approximations.

The alternative to computing the actual derivatives of the rotation matrix with respect to the axis-angle vector is either the use of finite differences or incremental rotations with analytical derivatives at the identity. Suppose, for example, that we are attempting to optimize the parameters of a rotation matrix in the context of an iterative method. The idea is to

replace the rotation matrix R at step k with another rotation R' given by the product of the current estimate and a perturbing rotation matrix $\exp([u]_{\times})$ which is initially equal to the identity. Thus, instead of taking the actual derivative of R with respect to its current axis-angle parameters as given in Eq. 15, the much simpler derivative of R' in terms of u is taken at the origin:

$$\frac{\partial R'}{\partial u_i} = R \left. \frac{\partial \exp([u]_{\times})}{\partial u_i} \right|_{u_i=0} = R G_i \tag{17}$$

The workaround of Eq. 17 found early advocates such as Taylor [60], or Drummond and Cipolla [18] and is popular in practice [19,33]. On the other hand, by all accounts, it is not the actual derivative of the rotation in terms of the axis-angle parameters, but rather the derivative of an expression that has a *corrective effect* on the current estimate. As such, it is a mapping that has the characteristics of a *retraction* from the tangent space of R to $\mathcal{SO}(3)$ as defined by Absil et al. [1] and can therefore be used to provide a valid direction of descent on the rotation manifold. The latter suggests that the application scope of this approach is limited only to problems involving iterative optimization.

A special class of problems which involve the recovery of orientation and have attracted considerable attention recently is that of *rotation averaging* [10,12,24,25]. The objective in this case is to recover the absolute or relative orientation most consistent with many estimates. Typically, these problems are solved iteratively and require the determination of a direction of descent either in parameter space or directly on the rotation manifold. Several solutions to rotation averaging problems presented by Hartley [25] incorporate the incremental rotation approach of Eq. 17 to establish descent directions.

Another approach for obtaining the derivatives of a rotation matrix is to parametrize it with a unit quaternion. Although practical by virtue of the rotation matrix expression in terms of a quaternion in Eq. 8, this approach unfortunately requires imposing a hard unit-norm constraint on the quaternion components. To impose this constraint in the context of Euclidean bundle adjustment, Lourakis and Argyros [37] optimize only the vector part v of a quaternion and implicitly obtain the scalar part as $\rho = \sqrt{1 - v^T v}$ (cf. Eq. 7). Clearly, this does not allow for negative scalar parts and, therefore, the rotation must be limited to the range $[-\pi/2, \pi/2]$. To overcome this, the initial orientation of every camera before the optimization is retained and only the difference from the initial orientation is optimized. This local update is certain to lie within the aforementioned range, and the approach is also referred to as an *incremental update* in Sect. 6.2.2 of [59].

4 Derivation of Modified Rodrigues Parameters with Stereographic Projection

Stereographic projection is a rational, bijective mapping from a sphere to a plane often encountered in complex analysis, topology, quantum computing, etc. However, it has been largely overlooked in the computer graphics and vision communities as a practical means of parameterizing orientation. In contrast, aerospace engineering literature has several notable references to the potential of this formalism, also known as the modified Rodrigues parameters (MRPs), e.g., [40,50,58,63,68].

4.1 Derivation of Projection/Back-Projection Maps

To establish notation for the rest of the paper, a brief derivation of the stereographic projection formulas is in order at this point. Consider a unit quaternion $q = \rho + v^T \varphi$ such that $\rho \in \mathbb{R}$ and $v \in \mathbb{R}^3$ with $\rho^2 + v^T v = 1$. We designate the “South Pole” of the sphere to be the unit quaternion $S = -1$. Let now $r(t)$ be the ray parameterized by $t \in \mathbb{R}$ passing through q and a purely imaginary quaternion $\psi^T \varphi$:

$$r(t) = S + t (\psi^T \varphi - S), \tag{18}$$

where $\psi \in \mathbb{R}^3$. Thus, the subspace of purely imaginary quaternions can be regarded as an equatorial hyperplane that “slices” the 4D unit sphere along 3 canonical directions through the origin and the “South Pole” as the center of projection, through which, the unit quaternion q is projected onto $\psi^T \varphi$ in the hyperplane. A visualization of this projection is provided in Fig. 1.

When the ray intersects the sphere, the resulting quaternions should have a unit norm, i.e., $|r(t)|^2 = 1$. Substituting the expression of $r(t)$ from Eq. 18 into the unit-norm constraint yields a solution for the parameter t when the ray intersects q :

$$t = \frac{2}{1 + \|\psi\|^2} \tag{19}$$

The unit quaternion can now be expressed in terms of ψ by substituting Eq. 19 into Eq. 18:

$$q = \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} + \frac{2}{1 + \|\psi\|^2} \psi^T \varphi \tag{20}$$

Conversely, it is fairly easy to project a unit quaternion onto the equatorial hyperplane. It suffices to solve first for $\|\psi\|^2$ in terms of the quaternion scalar part, ρ :

$$\|\psi\|^2 = \frac{1 - \rho}{1 + \rho} \tag{21}$$

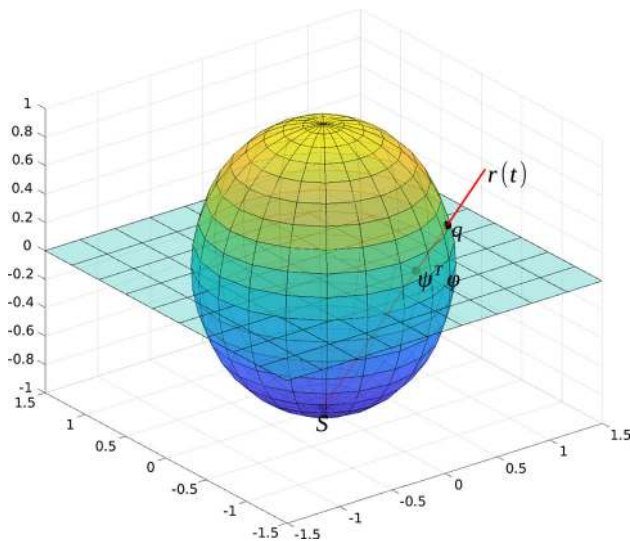


Fig. 1 A visualization of stereographic projection in 3D. The unit quaternion $S = -1$ is the center of projection and $\psi^T \phi$ is a quaternion in the equatorial plane. The ray $r(t) = S + t(\psi^T \phi - S)$ intersects the unit sphere at q

Thus, ψ can be expressed in terms of the components of q using Eqs. 20 and 21:

$$\psi = \frac{v}{1 + \rho} \quad (22)$$

It should be stressed here that the components of ψ can assume infinite values. Thus, more rigorously, $\psi \in \overline{\mathbb{R}}^3$, where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the *affinely extended set of real numbers* and this notation will be used throughout the rest of this paper.

4.2 Related Parameterizations: The Gibbs Vector

For completeness, we remark that a representation closely related to MRPs is the *Gibbs* or *classical Rodrigues* parameter vector g [25, 51, 58]. The Gibbs vector is defined by the projection of the quaternion parameters from the center of the unit sphere onto the hyperplane tangent to its “South Pole”, given algebraically as:

$$g = \frac{v}{\rho} \quad (23)$$

Classical Rodrigues parameters provide a minimal DoF representation that is singular and discontinuous at the angle of rotation π . Using the axis-angle parameterization $\rho = \cos(\frac{\theta}{2})$ and $v = \frac{\sin(\frac{\theta}{2})}{\theta} \omega$, it is straightforward to derive a relationship between g and ω [58]:

$$g = \frac{\tan \frac{\theta}{2}}{\theta} \omega \quad (24)$$

As demonstrated in Sect. 5, the relationship between the Gibbs vector and the axis-angle vector given in Eq. 24 becomes particularly useful in deriving a Cayley transform from the space of MRP vectors to the respective rotation matrices.

5 Modified Rodrigues Parameters as Vectors Parallel to the Rotation Axis

Depending on the choice of projection center, the resulting expressions for the coordinates of the projected quaternion on the equatorial plane will vary. In order for these coordinates to be valid modified Rodrigues parameters, the projection center should lie on the real axis⁴ as is the case with the derivation of Sect. 4. To state this more clearly, consider a unit quaternion $q = \rho + v^T \phi$ where $\phi = [i \ j \ k]^T$ and its axis-angle parameterization, such that:

$$\rho = \cos \frac{\theta}{2}$$

$$v = \frac{\sin \frac{\theta}{2}}{\theta} \omega$$

with $\omega \in \mathbb{R}^3$ and $\|\omega\| = \theta$. The vector of modified Rodrigues parameters associated with q is the triplet of stereographic coordinates ψ that back-projects to the corresponding spherical point in the following way:

$$\frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} = \rho = \cos \frac{\theta}{2} \quad (25)$$

$$\frac{2\psi}{1 + \|\psi\|^2} = v = \frac{\sin \frac{\theta}{2}}{\theta} \omega \quad (26)$$

Thus, the vector part of the quaternion is always represented by $2\psi/(1 + \|\psi\|^2)$, which is collinear with the parameter vector.

With ψ being parallel to v , it follows from Eq. 26 that it is also parallel to the rotation axis. This implies that MRPs are a member of the so-called family of *vectorial parameterizations* [4]. In particular, MRPs and Gibbs vectors belong to the *tangent* family, which enjoys certain important properties, the most prominent of them being the inter-connections in terms of the Cayley transform explained in Sect. 5.1. Using Eqs. 25 and 26, the relationship between the axis-angle vector ω and the MRPs of a rotation is straightforward for a rotation angle $\theta \in [0, 2\pi)$ [40, 58]:

$$\psi = \frac{\tan \frac{\theta}{4}}{\theta} \omega \quad (27)$$

⁴ In other words, it should be either 1 or -1 .

Thus, comparing Eq. 27 to the corresponding relationship for Gibbs vectors in Eq. 24, it can be inferred that the MRP vector ψ has twice the rotational range of the classical Rodrigues parameters g . In the special case where $\theta = 2\pi$, any MRP vector ψ with at least one of its coordinates equal to “infinity” will back-project to -1 on the quaternion sphere. More details on this representation peculiarity are given in Sect. 5.2.

Furthermore, using Eqs. 26 and 25, it is easy to derive a composition rule between MRP vectors based on quaternion multiplication. Specifically, for the unit quaternion product $q_3 = q_1q_2$, we obtain the following corresponding relationship in MRP space [58]:

$$\psi_3 = \frac{(1 - \|\psi_2\|^2)\psi_1 + (1 - \|\psi_1\|^2)\psi_2 - 2\psi_1 \times \psi_2}{1 + \|\psi_1\|^2\|\psi_2\|^2 - 2\psi_1 \cdot \psi_2} \tag{28}$$

5.1 MRPs and the Cayley Transform

The Cayley transform is a mapping from the space of skew-symmetric matrices directly to the group of rotation matrices. In particular, it is straightforward to show that the Cayley transform maps the skew symmetric matrix $[g]_\times$ of a Gibbs vector to the respective rotation matrix as follows [40,62]:

$$R = (I_3 + [g]_\times)(I_3 - [g]_\times)^{-1} \tag{29}$$

where I_3 is the 3×3 identity matrix. It is relatively easy to show that both factors in the product commute and are invertible⁵ for any skew-symmetric matrix $[g]_\times$. The inverse Cayley transform maps a rotation matrix to its corresponding Gibbs vector as follows:

$$[g]_\times = (R - I_3)(R + I_3)^{-1} \tag{30}$$

The transformation from an MRP skew-symmetric matrix $[\psi]_\times$ to the corresponding rotation matrix is a so-called second-order Cayley transform, given by the following mapping [62]:

$$R = (I_3 + [\psi]_\times)^2 (I_3 - [\psi]_\times)^{-2} \tag{31}$$

It is worth outlining here the rationale behind the derivation of the Cayley transform for MRPs as expounded by Schaub et al. [52]. The idea is to observe the relationship between classical Rodrigues parameters (Gibbs vectors) and MRPs through axis-angle vectors as given in Eqs. 24 and 27. Evidently, the rotation matrix obtained by employing the first-order (standard) Cayley transform formula of Eq. 29 on $[\psi]_\times$ yields a

rotation matrix with half the angle of the actual rotation that corresponds to the MRP triplet. This can be demonstrated with the aid of exponential notation:

$$\exp\left(\frac{1}{2}[\omega]_\times\right) = (I_3 + [\psi]_\times)(I_3 - [\psi]_\times)^{-1}, \tag{32}$$

where $R = \exp([\omega]_\times)$. Thus, since matrices $I_3 - [\omega]_\times$ and $(I_3 - [\omega]_\times)^{-1}$ commute, squaring both sides in Eq. 32 yields the second-order Cayley transform for MRPs given in Eq. 31. Schaub et al. report that, unlike classical Rodrigues parameters, there exists no similar expression for the inverse mapping from rotation matrices to MRPs [52].

5.2 Negated Quaternions and Shadow MRPs

Two antipodal (i.e., opposite) quaternions q and $-q$ represent the same rotation. Nevertheless, the stereographic coordinates of these two quaternions will differ. Markley calls the stereographic coordinates of the negated quaternion *shadow MRPs* [40]. Shadow coordinates can be easily worked-out from Eq. 20:

$$\eta = -\frac{\psi}{\|\psi\|^2} \tag{33}$$

It should be noted that the last formula is not valid for $\psi = (0, 0, 0)$, in which case the shadow quaternion coincides with the center of projection (i.e., the chosen “South Pole”) and the rotation has a single representation at the origin of the hyperplane. As the MRP vector moves far away from the origin of the hyperplane, the corresponding quaternions converge asymptotically to $q = -1$ at the projection center. This is a direct consequence of the fact that the line $y = -1$ is the asymptote at both $+\infty$ and $-\infty$ of function $(1-t^2)/(1+t^2)$, i.e., the quaternion’s scalar part from Eq. 25, as shown in Fig. 2. The plot clearly indicates that decay is fast and for $|t| \geq 15$, the function is already very close to -1 ; it becomes even closer for larger values. For example, the deviation of the function from -1 for $t = 10^2$ is in the order of 10^{-4} .

The above observation suggests that it is possible to approach the projection center with high accuracy using MRP vectors whose components are well within the nominal floating point range. Hence, this is a reasonable alternative to representing the quaternion at the center of projection without resorting to shadow coordinates which could cause discontinuities in applications such as interpolation. However, as will be explained in the following sections, our analysis is focused on the quaternion sphere, while MRPs are essentially used to provide the theoretical underpinning that allows us to work with unit quaternions without the need to impose the norm constraint. Thus, with the exception of interpola-

⁵ One way of showing this is to observe that matrices $I_3 + [\psi]_\times$ and $I_3 - [\psi]_\times$ have same eigenvectors and non-vanishing complex conjugate corresponding eigenvalues.

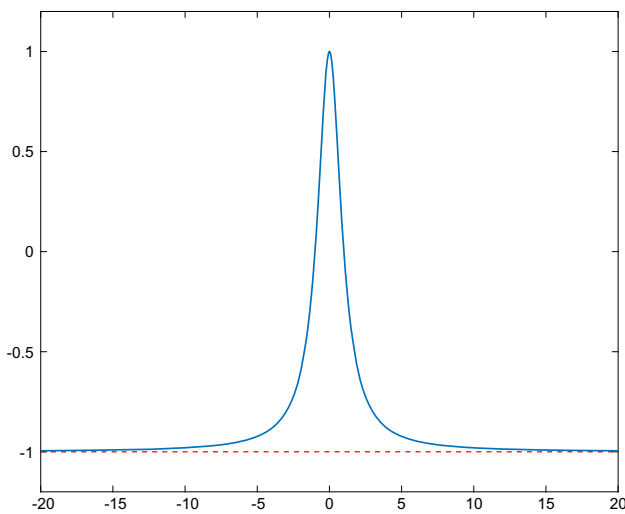


Fig. 2 Plot of function $(1 - t^2)/(1 + t^2)$ for $|t| \leq 20$ shown in solid blue. The horizontal asymptote $y = -1$ is shown with a red dashed line (Color figure online)

tion, unit quaternions can be directly manipulated in terms of their MRPs, without having to explicitly switch parameter domain.

6 Differential Properties of Unit Quaternions with Respect to Modified Rodrigues Parameters

It is clear from Eqs. 26, 25 that the derivatives of a unit quaternion with respect to its MRPs are rational functions of ψ . This is an advantageous fact not only from the aspect of computational efficiency but, most importantly, in terms of the complexity of the resulting expressions in the Jacobian [61]. In this section, we will provide a very important novel observation regarding these derivatives, which will not only simplify the computation of the Jacobian of a rotation matrix (refer to “Appendix A” for a complete set of formulas), but will also provide simple relationships between the derivatives of curves on the hyperplane and the derivatives of their back-projections on the sphere.

6.1 Quaternion Jacobian

Proposition 1 Let $q = \rho + v^T \varphi$ where $\varphi = [i \ j \ k]^T$ be a unit quaternion. Then, the Jacobian ∇q of q with respect to its modified Rodrigues parameters is:

$$\nabla q = \nabla \begin{bmatrix} v \\ \rho \end{bmatrix} = - \begin{bmatrix} vv^T - (1 + \rho) I_3 \\ (1 + \rho) v^T \end{bmatrix} \quad (34)$$

Proof Let $\psi \in \mathbb{R}^3$ be the MRPs of the unit quaternion. It follows from Eq. 26 that the derivative of the vector part v with respect to ψ will be:

$$\begin{aligned} \frac{\partial v}{\partial \psi} &= \frac{\partial \frac{2\psi}{1 + \|\psi\|^2}}{\partial \psi} = - \frac{2\psi \frac{\partial (1 + \|\psi\|^2)}{\partial \psi}}{(1 + \|\psi\|^2)^2} + \frac{2}{1 + \|\psi\|^2} \frac{\partial \psi}{\partial \psi} \\ &= - \frac{2\psi 2\psi^T}{(1 + \|\psi\|^2)^2} + \frac{2}{1 + \|\psi\|^2} I_3 \\ &= - \left(\frac{2\psi}{1 + \|\psi\|^2} \right) \left(\frac{2\psi^T}{1 + \|\psi\|^2} \right) \\ &\quad + \frac{2 + \|\psi\|^2 - \|\psi\|^2}{1 + \|\psi\|^2} I_3 \\ &= - \left(\frac{2\psi}{1 + \|\psi\|^2} \right) \left(\frac{2\psi^T}{1 + \|\psi\|^2} \right) \\ &\quad + \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} \right) I_3 \\ &= -vv^T + (1 + \rho) I_3 \end{aligned}$$

Similarly, using Eq. 25, the derivative of the scalar part in terms of ψ will be:

$$\begin{aligned} \frac{\partial \rho}{\partial \psi} &= - \frac{(1 - \|\psi\|^2) \frac{\partial (1 + \|\psi\|^2)}{\partial \psi}}{(1 + \|\psi\|^2)^2} + \frac{\frac{\partial (1 - \|\psi\|^2)}{\partial \psi}}{1 + \|\psi\|^2} \\ &= \frac{\frac{\partial \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2}}{\partial \psi}}{\partial \psi} = - \frac{(1 - \|\psi\|^2) 2\psi^T}{(1 + \|\psi\|^2)^2} - \frac{2\psi^T}{1 + \|\psi\|^2} \\ &= - \left(\frac{2}{1 + \|\psi\|^2} \right) \left(\frac{2\psi^T}{1 + \|\psi\|^2} \right) \\ &= - \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} \right) \left(\frac{2\psi^T}{1 + \|\psi\|^2} \right) \\ &= - (1 + \rho) v^T \end{aligned}$$

The Jacobian formula of Eq. 34 is very important because it has low complexity of expressions and involves only the quaternion components in simple additions and multiplications without the need of additional constraints. This means that, by virtue of the rotation matrix expression in terms of a unit quaternion given in Eq. 8, the components of the rotation matrix Jacobian tensor will in turn comprise simple polynomial expressions of the quaternion components (see “Appendix A”). In other words, the computation of the rotation matrix derivatives entails exclusively multiplications and additions on previously stored quantities.

6.2 Quaternion Updates from Perturbations in MRPs

Although the Jacobian of a unit quaternion with respect to MRPs can be expressed without the explicit presence of the parameters in the respective expressions, it will, however, produce a perturbation in parameter space during iterative optimization. Thus, it would appear that, in order to obtain

the new quaternion estimate, one is required to convert it to MRPs, then perform the update in \mathbb{R}^3 and eventually, convert the resulting parameters to the new quaternion estimate. In Proposition 2, we show that the current estimate of a quaternion can be updated without alternating among parameter spaces.

Proposition 2 Consider a perturbation $\delta \in \mathbb{R}^3$ in the modified Rodrigues parameters of a unit quaternion $q = \rho + v^T \varphi$. Then, the scalar and vector part of the unit quaternion $q' = \rho' + v'^T \varphi$ corresponding to the perturbed MRPs can be obtained as follows:

$$v' = \frac{v + (1 + \rho) \delta}{1 + v \cdot \delta + \frac{1}{2} (1 + \rho) \|\delta\|^2} \tag{35}$$

$$\rho' = \frac{\rho - v \cdot \delta - \frac{1}{2} (1 + \rho) \|\delta\|^2}{1 + v \cdot \delta + \frac{1}{2} (1 + \rho) \|\delta\|^2} \tag{36}$$

Proof Let ψ be the MRP triplet associated with q . Then, taking the stereographic projection formula in Eq. 26 for v' , we have:

$$\begin{aligned} v' &= \frac{2(\psi + \delta)}{1 + \|\psi + \delta\|^2} = \frac{\frac{2(\psi + \delta)}{1 + \|\psi\|^2}}{\frac{1 + (\psi + \delta) \cdot (\psi + \delta)}{1 + \|\psi\|^2}} \\ &= \frac{\frac{2\psi}{1 + \|\psi\|^2} + \frac{2}{1 + \|\psi\|^2} \delta}{1 + \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta + \frac{1}{2} \left(\frac{2}{1 + \|\psi\|^2}\right) \|\delta\|^2} \\ &= \frac{\frac{2\psi}{1 + \|\psi\|^2} + \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2}\right) \delta}{1 + \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta + \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2}\right) \|\delta\|^2} \\ &= \frac{v + (1 + \rho) \delta}{1 + v \cdot \delta + \frac{(1 + \rho)}{2} \|\delta\|^2} \end{aligned}$$

Similarly, taking the stereographic projection formula in Eq. 25 for ρ' , yields:

$$\begin{aligned} \rho' &= \frac{1 - \|\psi + \delta\|^2}{1 + \|\psi + \delta\|^2} = \frac{\frac{1 - (\psi + \delta) \cdot (\psi + \delta)}{1 + \|\psi\|^2}}{\frac{1 + (\psi + \delta) \cdot (\psi + \delta)}{1 + \|\psi\|^2}} \\ &= \frac{\frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} - \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta - \frac{1}{2} \left(\frac{2}{1 + \|\psi\|^2}\right) \|\delta\|^2}{1 + \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta + \frac{1}{2} \left(\frac{2}{1 + \|\psi\|^2}\right) \|\delta\|^2} \\ &= \frac{\frac{1 - \|\psi\|^2}{1 + \|\psi\|^2} - \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta - \frac{1}{2} \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2}\right) \|\delta\|^2}{1 + \left(\frac{2\psi}{1 + \|\psi\|^2}\right) \cdot \delta + \frac{1}{2} \left(1 + \frac{1 - \|\psi\|^2}{1 + \|\psi\|^2}\right) \|\delta\|^2} \\ &= \frac{\rho - v \cdot \delta - \frac{1}{2} (1 + \rho) \|\delta\|^2}{1 + v \cdot \delta + \frac{1}{2} (1 + \rho) \|\delta\|^2} \end{aligned}$$

Propositions 1 and 2 have a significant impact in the way rotations parameterized with MRPs are updated during iterative optimization. In particular, Proposition 1 ensures that

the elements of the rotation matrix Jacobian are computed with a few multiplications and additions of previously stored numbers (i.e., the four quaternion components); furthermore, Proposition 2 ensures that the updated rotation matrix in each step of the iterative method can be obtained without having to compute the MRPs of the previous estimate. In other words, both the Jacobian computation as well as the update of the rotation matrix do not explicitly require the use of MRPs and both can be computed with a few primitive operations on previously stored numbers.

6.3 Arc Length of Quaternion Curves parameterized with MRPs

Being a rational map, stereographic projection can be used to back-project smooth 3D curves on the sphere. The resulting spherical curves have certain differential attributes which could be useful in manipulating their properties in the more familiar space \mathbb{R}^3 .

Lemma 1 For any unit quaternion q , the Gram matrix of the Jacobian with respect to its MRPs is a scalar multiple of the 3×3 identity matrix:

$$(\nabla q)^T \nabla q = (1 + \rho)^2 I_3 \tag{37}$$

Proof Using the result of Proposition 1, we have:

$$\begin{aligned} (\nabla q)^T \nabla q &= \begin{bmatrix} vv^T - (1 + \rho) I_3 \\ (1 + \rho) v^T \end{bmatrix}^T \begin{bmatrix} vv^T - (1 + \rho) I_3 \\ (1 + \rho) v^T \end{bmatrix} \\ &= [vv^T - (1 + \rho) I_3 \quad (1 + \rho) v] \begin{bmatrix} vv^T - (1 + \rho) I_3 \\ (1 + \rho) v^T \end{bmatrix} \\ &= (v^T v - 2(1 + \rho) + (1 + \rho)^2) vv^T + (1 + \rho)^2 I_3 \\ &= \left(\underbrace{\rho^2 + v^T v}_1 - 1 \right) vv^T + (1 + \rho)^2 I_3 \\ &= (1 + \rho)^2 I_3 \end{aligned}$$

A direct consequence of Lemma 1 is that the columns of the quaternion Jacobian must be orthogonal for every unit quaternion $q \neq -1$. More formally:

Corollary 1 For any unit quaternion $q \neq -1$, the columns of the Jacobian of q with respect to its MRPs constitute an orthogonal basis of the tangent space of the quaternion sphere at q .

Lemma 2 Let $\psi(t) = (x(t), y(t), z(t)) : \mathbb{R} \rightarrow \mathbb{R}^3$ be a smooth curve. Then the unit quaternion function $q(t) = \rho(t) + (v(t))^T \varphi$ obtained as the stereographic back-projection of $\psi(t)$ on the quaternion sphere is also smooth and the arc length $s(t)$ of $q(t)$ is given by the following expression:

$$s(t) = 2 \int_0^t \frac{1}{1 + \|\psi(u)\|^2} \left\| \frac{d\psi}{du} \right\| du \quad (38)$$

Proof It naturally follows from Proposition 1 that, by means of the chain rule, the first derivative will be a product of smooth functions and consequently, higher order derivatives will be the sum of such products. Thus, the spherical back-projection of $\psi(t)$ will be smooth. Another way of arriving at this conclusion is to simply consider that stereographic projection is a rational mapping and therefore maps smooth functions to smooth functions.

To prove Eq. 38, we make use of Lemma 1, starting from the standard formula for the arc length of $q(t)$:

$$\begin{aligned} s(t) &= \int_0^t \sqrt{\left(\frac{dq}{du}\right)^T \frac{dq}{du}} du \\ &= \int_0^t \sqrt{\left(\nabla q \frac{d\psi}{du}\right)^T \left(\nabla q \frac{d\psi}{du}\right)} du \\ &= \int_0^t \sqrt{\left(\frac{d\psi}{du}\right)^T \underbrace{(\nabla q)^T \nabla q}_{=(1+\rho(u))^2 I_3} \frac{d\psi}{du}} du \\ &= \int_0^t (1 + \rho(u)) \left\| \frac{d\psi}{du} \right\| du \end{aligned}$$

It can be easily inferred from Eq. 25 that $1 + \rho(u) = \frac{2}{1 + \|\psi(u)\|^2}$. Thus,

$$s(t) = \int_0^t \frac{2}{1 + \|\psi(u)\|^2} \left\| \frac{d\psi}{du} \right\| du$$

7 Quaternion Interpolation

Quaternion interpolation is ubiquitous in the fields of computer graphics, robotics and aerospace engineering [8, 21, 22, 44, 55]. Generating smooth orientation paths between key orientations is a very challenging task, primarily because we wish to attach linear interfaces onto steering mechanisms which, by definition, manipulate objects (i.e., rotations) that reside in a spherical manifold. In other words, the desired attributes of the generated sequences are hard to attain, due to the topological nonlinearities of the group of rotations.

Stereographic projection is a smooth, bijective mapping from \mathbb{R}^3 to the unit sphere in \mathbb{R}^4 . Consequently, lines in 3D become distorted on the sphere to account for the incompatibility between the two topological spaces. In this section, we provide a simple solution to cope with the distortion caused by perspective projection, while designing the spherical curve in the hyperplane. The idea is to interpolate the derivative of the spherical curve, in addition to interpolating the data.

7.1 Configuring Unit Quaternion Derivatives on the Hyperplane

Suppose we wish to establish a relationship between the derivative of a parametric unit quaternion curve $q(t) = \rho(t) + (v(t))^T \varphi$ and the derivative of the corresponding MRP curve $\psi(t)$. The chain rule for $q(t)$ yields:

$$q'(t) = \nabla q \psi'(t), \quad (39)$$

where $\psi'(t) = \frac{d\psi}{dt}$ and ∇q is the Jacobian of the quaternion curve at ψ . Clearly, Eq. 39 defines a 4×3 linear system of equations in the components of $\psi'(t)$. We claim that for $q(t) \neq -1$ this system has the following unique solution:

$$\psi'(t) = \frac{1}{(1 + \rho)^2} (\nabla q)^T q'(t) \quad (40)$$

Lemma 3 For the unit quaternion $q = \rho + v^T \varphi$, examine the 4×3 linear system $\nabla q \xi = b$, where $b \in \mathbb{R}^4$ and ∇q is the Jacobian of q with respect to its modified Rodrigues parameters.

(a) For $q \neq -1$, consider the vector:

$$\xi = \frac{1}{(1 + \rho)^2} (\nabla q)^T b \quad (41)$$

- (i) If b lies in the tangent space of q , then ξ is the unique solution of the system.
- (ii) if b is not in the tangent space of q , then ξ is the least squares minimizer of the system and consequently, $\nabla q \xi$ is the projection of b on the tangent space of q .

(b) For $q = -1$:

- (i) If $b \neq 0$, the system has no solutions.
- (ii) If $b = 0$, any $\xi \in \mathbb{R}^3$ is a solution.

Proof The proof is trivial for $q = -1$. For $q \neq -1$, we multiply by ∇q on the left to get the 3×3 equivalent system:

$$\left((\nabla q)^T \nabla q \right) \xi = (\nabla q)^T b$$

From Lemma 1, we know that the Gram matrix of the gradient will be a non-zero scalar multiple of the identity:

$$\begin{aligned} (1 + \rho)^2 \xi &= (\nabla q)^T b \\ \Leftrightarrow \xi &= \frac{1}{(1 + \rho)^2} (\nabla q)^T b \end{aligned}$$

The solution of the 3×3 overdetermined system in Eq. 41 will satisfy all the original equations because b is in the tangent

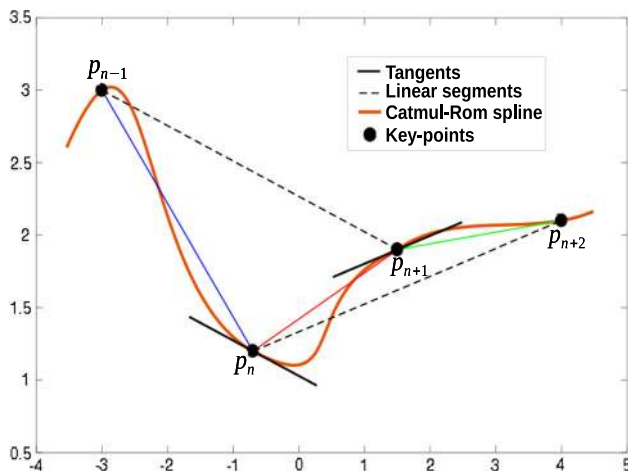


Fig. 3 Standard Catmull–Rom interpolation. The tangents at the data points p_n and p_{n+1} (shown as black solid lines) are parallel to the linear segments defined by p_{n-1}, p_{n+1} and p_n, p_{n+2} (shown with dashed lines)

space of q and we know from Corollary 1 that it can be expressed as a linear combination of the columns of ∇q .

7.2 Spherical Catmull–Rom Splines manufactured in 3D

Equation 41 provides the means to configure the differential properties of a 4D spherical curve by manipulating its projection in \mathbb{R}^3 . To demonstrate how effective this approach is in terms of eliminating projective distortion, we present a sample scheme for designing spherical Catmull–Rom splines [9].

Standard Catmull–Rom splines are composed of polynomial segments with end-point derivatives that match the slope of the linear segments that connect data points immediately preceding and trailing the end-points (see Fig. 3). We consider an analog of Catmull–Rom splines on the sphere in which we require the derivative of the spherical curve at a specific data-point to be collinear to the tangent-space projection of the linear segment defined by the trailing and preceding data points (Fig. 4).

To interpolate between two key points q_n and q_{n+1} according to the spherical analog of Catmull–Rom splines described in this section, we must solve for the coefficients of a cubic polynomial $\psi(t) = b_3t^3 + b_2t^2 + b_1t + b_0$, where $b_0, b_1, b_2, b_3 \in \mathbb{R}^3$ and $t \in [0, 1]$. It follows that $\psi(t)$ should interpolate the projections of two data points at ψ_n and ψ_{n+1} for $t = 0$ and $t = 1$, respectively. Hence, the following conditions should apply:

$$b_0 = \psi_n \tag{42}$$

$$b_3 + b_2 + b_1 + b_0 = \psi_{n+1} \tag{43}$$

Furthermore, we require that the tangents at the spherical points are parallel to the chords that connect the trailing and

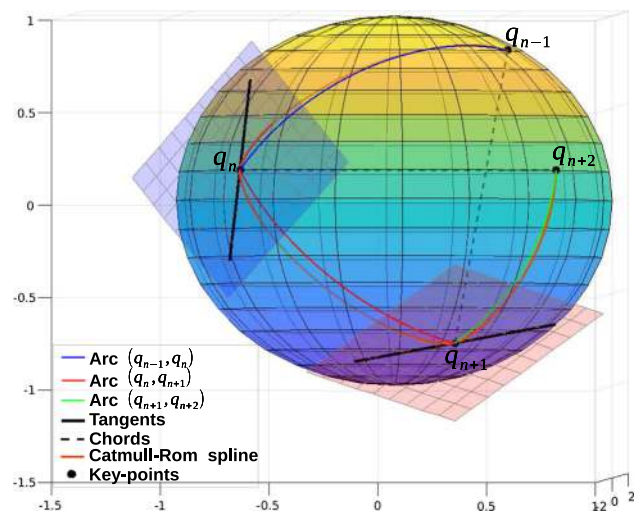


Fig. 4 Spherical Catmull–Rom interpolation. The tangents of the spherical curve at q_n and q_{n+1} are parallel to the projections of linear segments $q_{n-1} q_{n+1}$ and $q_n q_{n+2}$ onto the tangent spaces of q_n and q_{n+1} , respectively

preceding data points, as shown in Fig. 4. Thus, two additional constraints are obtained on the coefficients of the cubic:

$$b_1 = \frac{\lambda (\nabla q_n)^T (q_{n+1} - q_{n-1})}{(1 + \rho_n)^2} \tag{44}$$

$$3b_3 + 2b_2 + b_1 = \frac{\lambda (\nabla q_{n+1})^T (q_{n+2} - q_n)}{(1 + \rho_{n+1})^2}, \tag{45}$$

where ρ_i is the scalar part of q_i and λ is a user-defined positive scalar. Using Eqs. 42, 43, 44 and 45, the coefficients of $\psi(t)$ can be computed in the following order:

$$\tau_1 = \frac{(\nabla q_n)^T (q_{n+1} - q_{n-1})}{(1 + \rho_n)^2} \tag{46}$$

$$\tau_2 = \frac{(\nabla q_{n+1})^T (q_{n+2} - q_n)}{(1 + \rho_{n+1})^2} \tag{47}$$

$$b_0 = \psi_n \tag{48}$$

$$b_1 = \lambda \tau_1 \tag{49}$$

$$b_3 = \lambda \tau_2 + b_1 - 2(\psi_{n+1} - b_0) \tag{50}$$

$$b_2 = \psi_{n+1} - b_3 - b_1 - b_0 \tag{51}$$

It should be noted that the scheme for designing spherical Catmull–Rom splines is intended as an example of a more general methodology for eliminating perspective distortion by configuring the planar curve to produce a back-projection that matches the desired differential properties on the sphere. The result of Eq. 41 is a tool with multiple uses when designing spherical splines on the plane.

Figure 5 illustrates an 8-point spherical Catmull–Rom spline along with the corresponding spherical quadrangle interpolation (SQUAD) curve. Clearly, both curves fulfill the

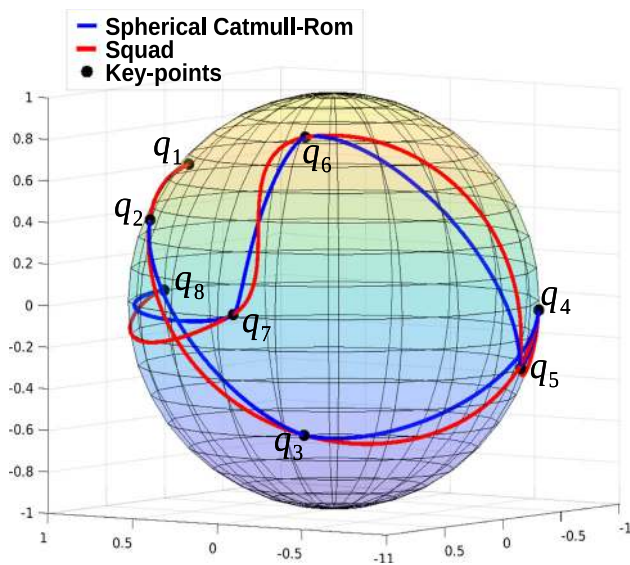


Fig. 5 Spherical Catmull–Rom (blue) and SQUAD interpolation (red) for 8 key-quaterrions (black dots) (Color figure online)

primary requirement of smoothness. Still, a more careful observation reveals that the spherical Catmull–Rom spline achieves a generally shorter distance between the key points in comparison with SQUAD. Obtaining shorter arc lengths in orientation interpolants is important in robotic and animation applications [14,27,49,53] because they yield reduced torque in the rotational motion. For similar reasons, minimal curvature is an equally important attribute [13]. Particularly in the case of animation, approximately constant speed in spherical curves is desirable because it can be warped into any desirable acceleration profile (e.g., trapezoidal moves) [66]. In summary, it would appear that connected great arcs can be loosely regarded as the “ideal” interpolants in terms of the aforementioned attributes (minimal arc length, minimal curvature and constant speed) if we could somehow overlook the lack of smoothness at the end-points. In Sect. 8.1, we study the characteristics of generated Catmull–Rom and SQUAD

curves and use the great arcs between data points as a benchmark to obtain quantifiable measures on their performance.

8 Experimental Results

This section presents experimental results comparing the performance of MRPs in various applications against alternative parameterizations of rotation.

8.1 Spherical Catmull–Rom Splines as Orientation Interpolants

The method proposed in Sect. 7.2 for the design of spherical Catmull–Rom (SCR) splines was merely a demonstration of the ways that the properties of a spherical curve can be configured in the hyperplane. However, the resulting curve, although not optimal in all aspects, exhibits, besides smoothness, significant improvements in terms of arc length and curvature when compared to the one produced by SQUAD. Furthermore, it approximates the great arcs between the key points better than SQUAD in the majority of cases.

For demonstration, we performed comparisons on three sequences of eight key quaterrions each, generated by successive random “jumps” in polar coordinates. In particular, the angles of each jump are uniformly sampled from a specified range in degrees. The range from which the angles are sampled determines the density of the key points which in turn affects the behavior of the interpolants. To observe the qualitative characteristics of the spherical curves in datasets of varying sparsity, the angular jumps used in the three generated sequences were sampled from the intervals $[10^\circ, 40^\circ]$, $[10^\circ, 70^\circ]$ and $[10^\circ, 100^\circ]$, respectively. Figures 6, 7, 8 illustrate arc length, speed and distance from great arc for both spherical Catmull–Rom splines and SQUAD in the three aforementioned sequences.

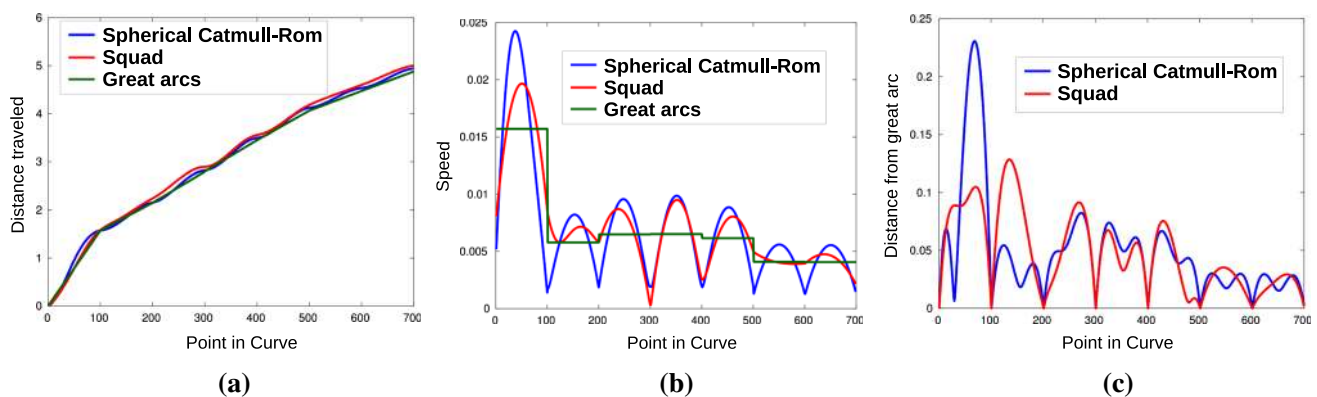


Fig. 6 Arc length, speed and distance from the great arcs for a sequence of 8 points generated using polar jumps in the interval $[10^\circ, 40^\circ]$ (to be viewed in color). **a** Arc length, **b** speed, **c** distance from great arc

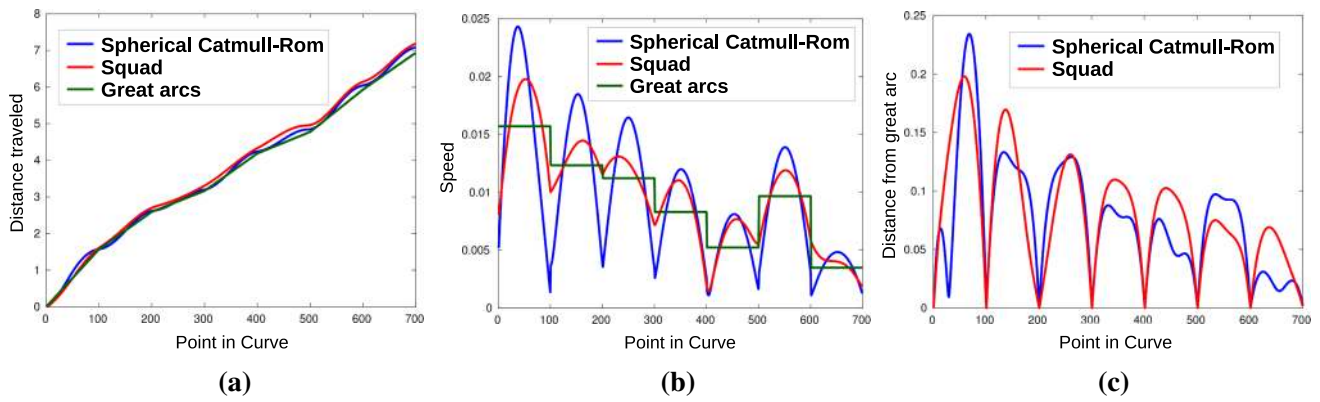


Fig. 7 Arc length, speed and distance from the great arcs for a sequence of 8 points generated using polar jumps in the interval $[10^\circ, 70^\circ]$. **a** Arc length, **b** speed, **c** distance from great arc

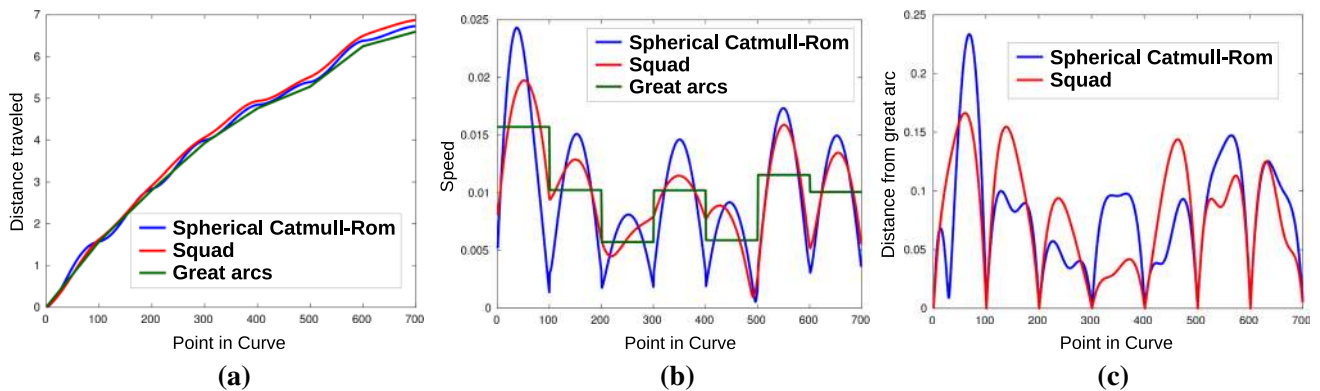


Fig. 8 Arc length, speed and distance from the great arcs for a sequence of 8 points generated using polar jumps in the interval $[10^\circ, 100^\circ]$. **a** Arc length, **b** speed, **c** distance from great arc

The results indicate that SCR splines produce curves that are often much more proximal to great arcs than those obtained from SQUAD, with generally shorter arc lengths, suggesting that the quality of interpolation attains similar, if not higher standards. The latter is an indication that perspective distortion is either not present or minimal as a direct consequence of configuring spherical derivatives in the hyperplane. On the other hand, SQUAD presents speed patterns that match SLERP closer than SCR splines. This is not necessarily a bad trait, primarily because SCR curves consistently produce a symmetric bell-shaped speed profile, which can be perceived as a smooth trapezoidal pattern. It should be noted that SQUAD is twice differentiable at the key points, while SCR curves are only continuous in the first derivative, a direct consequence of the definition of Catmull–Rom splines. In summary, SCR curves are smooth interpolants presenting little distortion on the sphere and bell-shaped symmetric speed patterns which can be easily warped to produce other speed profiles. On the other hand, velocity is not differentiable at the key points, a fact which may bear consequences, depending on the application. It should be noted, however, that the underlying general interpolation rationale involving the interpolation of derivatives side-to-side with data points is a method successful in

producing spherical interpolants which are devoid of distortion and can be adapted to suit spherical interpolation based on more general planar curves such as B-splines, for instance.

8.2 Descent Behavior of MRPs

Aiming to assess the descent behavior of MRPs in a situation involving a single unknown rotation, we chose to employ *absolute orientation* [29] as a test problem. It is widely known that this problem can be dealt with non-iteratively, e.g., [15, 28, 29, 36]. However, our objective in this experiment was not to provide yet another solution, but rather to benchmark how MRPs compare against other parameterization schemes in the context of a basic, quadratic minimization problem in only the rotation parameters. Specifically, the parameterization schemes compared with MRPs were axis-angle, normalized quaternion and incremental rotation.

Given two $3 \times N$ matrices X and Y comprised of N corresponding points in two different reference frames with the same origin, absolute orientation requires determining the rotation $R(p^*)$, where

$$p^* = \arg \min_p \|R(p) Y - X\|^2, \tag{52}$$

$p \in \mathbb{R}^3$ is the orientation parameter vector and $\|\cdot\|$ denotes the Frobenius norm for matrices.⁶

In our experimental setup, the dataset comprises 100 correspondences, i.e., matrices X and Y have size 3×100 . The unrotated points X were sampled from a 3D Gaussian with a covariance matrix $10^2 I_3$, thus producing a “spread” of roughly 10 metric units. The ground-truth rotation matrix was synthesized by uniformly sampling the 3 Euler angles from the interval $[20^\circ, 80^\circ]$. This rotation was then applied to every row of X to yield Y .

To study the descent behavior of MRPs against alternative parameterizations across multiple levels of noise, the optimization was carried out for 100 incremental standard deviation levels of noise from 0 to 2.5 using the Levenberg–Marquardt algorithm [34,42]. The noise is presumed Gaussian and was added to Y , which, by virtue of the property of linear propagation of covariance, is statistically equivalent to contaminating the relative position of the correspondences.

For each noise level, the same experiment (i.e., using the same ground-truth data) was repeated 40 times from a random starting point to convergence and the error in each step was recorded, as well as the overall steps to convergence (or maximum permissible iterations). Since we know that the absolute orientation cost function has 4 stationary points which are the eigenvectors of a data-dependent matrix, we would expect the process to occasionally get stuck in one of the 3 suboptimal points. In practice, we observed that this rarely happens. However, to ensure that the descent observations are not biased by the occasional convergence to local minima, we used median values for the error and the number of steps to convergence. Figure 9 illustrates plots of steps-to-convergence versus standard deviation of Gaussian noise. It should be noted that the iterative process terminates when any of the following conditions are met: (a) squared error below 10^{-6} , (b) change in squared error below 10^{-12} and, (c) the process has reached 100 iterations.

The incremental rotation approach employed here has been adopted by several notable pieces of work in vision, e.g., [18,33,39,60]. The rationale behind this approach is to take advantage of the fact that the tangent space of a rotation matrix R comprises all matrices RS_x where S_x is a skew-symmetric matrix. It is therefore possible to devise a very simple retraction [1] R' , which maps the tangent space of R onto $\mathcal{SO}(3)$ by applying a perturbing rotation on the right⁷ of R :

⁶ Solving absolute orientation amounts to estimating a rotation and a translation. Yet, as explained by Horn in [28], the problem can be reformulated to an equivalent one involving only rotation. Historically, this rotation-only formulation was originally introduced in astronautics as a satellite attitude estimation problem by Wahba [65].

⁷ Could also be applied on the left.

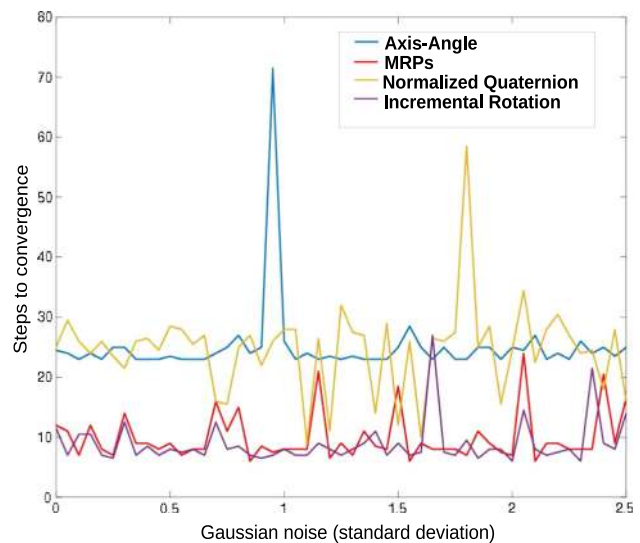


Fig. 9 Steps to convergence for added Gaussian noise up to 2.5 in a synthetic point-set with spread over 10 metric units

$$R'(S_x) = R \exp(S_x) \tag{53}$$

It can be easily shown that R' is a retraction, since the exponential map is smooth and $R'([0]_x) = R \exp([0]_x) = R$. Most importantly, R' satisfies the *local rigidity* requirement, since the directional derivative

$$\left. \frac{dR'(tU_x)}{dt} \right|_{t=0} = RU_x$$

is the identity mapping in the tangent space of R for any skew-symmetric matrix U_x . Consequently, the 3 descent directions on the manifold are RG_1, RG_2, RG_3 , and they are obtained by differentiating R' at the origin.

The results of Fig. 9 clearly indicate that MRPs and incremental rotations consistently reach a converged state within 10 to 20 iterations, while normalized quaternions and axis-angle parameters require roughly between 20 to 60 iterations. We conjecture that the incremental approach combined with MRPs would most likely improve the number of steps to convergence, but the gain would be marginal. This conjecture is based on the observation (using the formulas in “Appendix A”) that the derivatives of a rotation matrix with respect to MRPs at the origin are the scaled multiples of infinitesimal rotations, $4G_1, 4G_2, 4G_3$.

The evolution of error during the entire Gauss-Newton process is consistent with the convergence rates of Fig. 9. Indicative plots of how the error evolves throughout the Gauss-Newton process for three different added Gaussian noise levels (standard deviation 0, 1.5 & 2.5) are shown in Fig. 10. The error in each step is obtained as the median estimate of the error values in the same step across 20 distinct Gauss-Newton executions. It is clear that MRPs and incre-

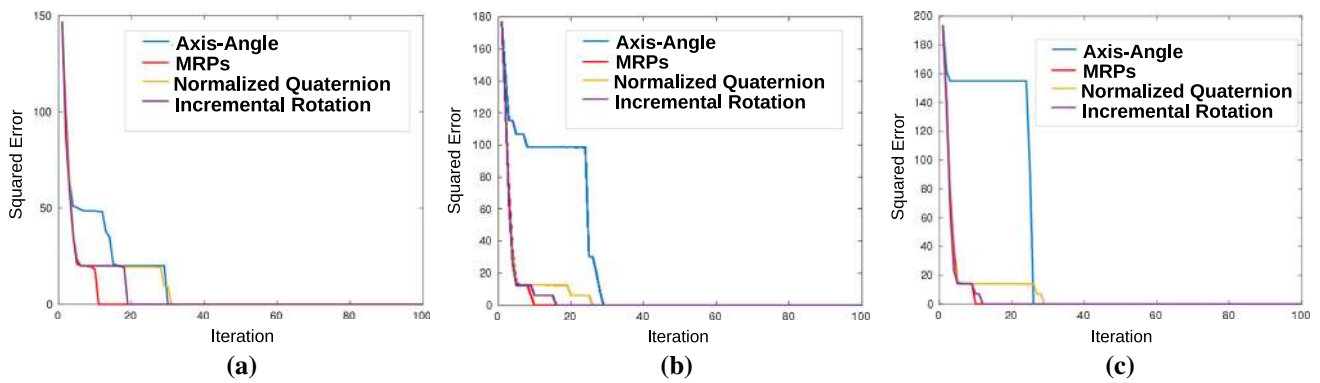


Fig. 10 Median error versus iteration in the Gauss-Newton method for added Gaussian noise with standard deviation 0, 1.5 and 2.5. **a** Noise level 0, **b** noise level 1.5, **c** noise level 2.5

mental rotations present a similar error curve with a very steep slope, as opposed to the axis-angle parameters and, to a lesser extent, normalized quaternion; furthermore, this pattern is consistent throughout the various levels of noise. We attribute this behavior primarily to the numerical stability of the rotation derivatives associated with the MRPs and the incremental rotation. In contrast, axis-angle and normalized quaternion Jacobians contain variable quantities in the denominator and very small/large values in these quantities may produce unstable descent patterns.

The noticeable abrupt “dives” in the error curves in Fig. 10 most likely correspond to periods in which the Levenberg–Marquardt damping factor grows upon successive steps in order to shorten the step size in the search for a better solution. This would account for the occasional nearly-constant error values for certain periods. Subsequently (but not in all cases), the error curve introduces a steep “dive” that corresponds to a step size that reached-out to an improved solution lying in a steeper region of the search surface.

8.3 Sparse Bundle Adjustment

Given a set of images depicting a number of 3D points from several different viewpoints, bundle adjustment (BA) is the problem of simultaneously refining the 3D coordinates of these points, as well as the parameters of the relative motion and possibly the optical characteristics of the camera(s) employed to acquire the images, according to an optimality criterion involving the cumulative image reprojection error of all points. BA amounts to a large, nonlinear optimization problem on the 3D structure and viewing parameters (i.e., camera pose and possibly intrinsic calibration and radial distortion). It is employed as the last step of most feature-based 3D reconstruction pipelines, since its solution yields a reconstruction which is optimal in the MLE sense under the assumption that the noise pertaining to the observed image features is zero-mean Gaussian [37].

BA constitutes a special type of a nonlinear least squares problem, since the lack of interaction among parameters for different 3D points and cameras results in the underlying normal equations having a special “arrowhead” sparse block structure [35]. *sba* [37] is a software package that efficiently solves BA using a sparse variant of the Levenberg–Marquardt algorithm that exploits the particular zero pattern of the underlying normal equations.

With the aid of publicly available, real-world datasets from [2], we tested different rotation parameterizations applied to BA. These datasets originate from incremental 3D reconstruction for large-scale community photo collections. More specifically, we used the first two data files from each of the “Ladybug”, “Trafalgar Square”, “Dubrovnik”, “Venice” and “Final” datasets. The *sba* [37] package was used to optimize those datasets using its default, quaternion-based local rotation parameterization described in Sect. 3.2. We also adapted *sba* to employ a global rotation parameterization based on the MRPs and compared it against the local quaternion parameterization. The results of the comparison are illustrated in Fig. 11, which shows the execution times and the number of iterations for the two rotation parameterizations.

In all applications of *sba*, the default convergence parameters were employed; in particular, the maximum number of iterations was set to 150. With the exception of the last data file (namely ‘394-100368’), both parameterizations converged to the same global minimum. In the last dataset, the parameterization employing MRPs required roughly four times more iterations but converged to a better minimum, which corresponded to over 60% lower average reprojection error compared to that obtained with quaternions. For the majority of datasets, both parameterizations required very similar numbers of iterations to converge. However, MRPs converged with noticeably fewer iterations for two datasets. The execution times were generally lower for the parameterization based on MRPs, owing to the simpler calculations involved in the evaluation of the image projections and their

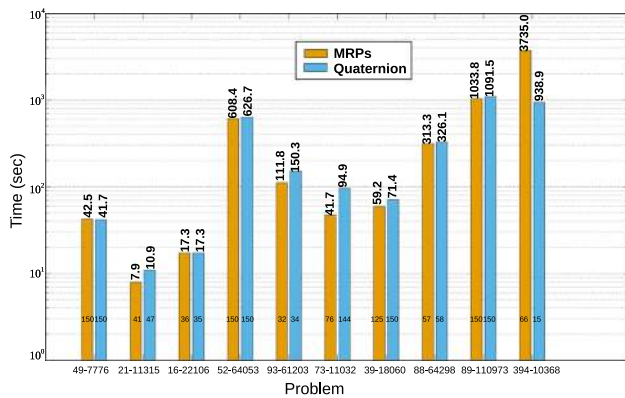


Fig. 11 Execution times for various BA datasets and two rotation parameterizations (orange for MRPs and blue for quaternions). Notice the logarithmic scale in the vertical axis. Following [2], each dataset in the horizontal axis is labeled as C - P , C and P being, respectively, the numbers of cameras and 3D points it involves. The actual execution times are shown in bold near the top of every bar. The numbers in italics near the bar bottoms are the iterations needed for convergence. The increased execution time required by MRPs for the rightmost dataset is due to that they required 4 times more iterations but converged to a much better minimum (Color figure online)

derivatives. Still, we note that the execution time for each iteration of `sba` is dominated by the time needed for the linear algebra operations involved in the solution of the normal equations (in particular, the Cholesky factorization of the Schur complement, cf. [37]), which does not depend on the choice of rotation parameterization.

8.4 Exterior Orientation

Exterior orientation refers to the estimation of the position and orientation of a camera given its intrinsic parameters and a set of $n \geq 3$ correspondences between known 3D points and their 2D image projections. This problem, also known as the PnP or camera resection problem,⁸ has received much attention due to its applicability in various domains. Exterior orientation is typically dealt with by embedding minimal-size PnP solvers to robust regression frameworks such as RANSAC (see [38] and references therein). However, as minimal solutions ignore much of the redundancy present in the data, they suffer from inaccuracies. To remedy this, an additional step comprised of nonlinear optimization with the Levenberg–Marquardt algorithm is employed to minimize the reprojection error pertaining to all inliers [38].

Starting with the datasets employed for bundle adjustment in Sect. 8.3, we extracted the 3D points projecting to their first frames. Then, those 3D points along with their projections

⁸ Strictly speaking, camera resectioning is slightly different since photogrammetrists define it as determining the projection matrix corresponding to a set of 3D–2D correspondences, i.e., the camera intrinsics are unknown.

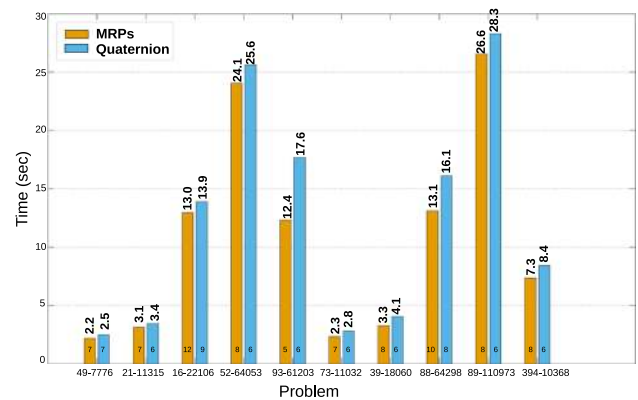


Fig. 12 Execution times for exterior orientation problems corresponding to the first frames of the datasets employed in Sect. 8.3, using two rotation parameterizations (orange for MRPs and blue for axis-angle). The actual execution times are shown in bold near the top of every bar, whereas the number of iterations are in italics at the bar bottoms (Color figure online)

in the first frame of each dataset were used for estimating the corresponding camera poses using the `posest` library implementing [38]. We also modified the nonlinear refinement step of `posest` to employ a rotation parameterization based on MRPs and compared it with its native axis-angle parameterization. Since the execution times for pose estimation are in most cases very small, and in order to accurately measure them, each pose estimation was run 100 times and the elapsed time scaled accordingly.

Figure 12 shows the execution times and the number of iterations for both rotation parameterizations. Similarly to the BA experiment described above, all optimizations converged to the same poses for both parameterizations. However, the execution times pertaining to MRPs are shorter, despite that the number of iterations is occasionally slightly higher compared to those spent for the exponential parameterization. Compared to the BA experiment, the difference between the execution time performance of the two parameterizations is more evident. This is due to the small size of the nonlinear minimization of the single view reprojection error and its consequent low computational cost, and clearly demonstrates the performance benefits gained by the use of MRPs.

9 Conclusion

Modified Rodrigues parameters is a formalism for the representation of orientation based on stereographic projection, originally introduced in the field of aerospace engineering by Wiener [68] in 1962. Stereographic projection is a well-established mathematical construct with primarily theoretical applications in complex analysis, topology and projective geometry. However, the practical significance of this mapping in applied fields such as computer vision, graphics and robotics has been overlooked.

This paper has advocated the use of MRPs for parameterizing rotations in problems arising in the fields of computer graphics and vision. Its primary objective is to familiarize the community with this formalism from the aspect of practical applications involving the recovery and/or interpolation of orientation by emphasizing its graceful properties not only as a rational parameterization but also in terms of differentiation.

In particular, it was shown that the Jacobian of a quaternion is not only a rational function of its MRPs, but also a polynomial function of its scalar and vector part. This is favorable from the perspective of nonlinear optimization problems involving the recovery of orientation, considering that the Jacobians corresponding to parametric unknowns such as normalized quaternions or axis-angle vectors yield occasionally highly complicated and non-rational expressions. In addition to the succinct Jacobian, it was shown that the update of a quaternion from a perturbation in its MRPs does not require the use of the actual parameter vector. This means that there is no need to move through parameter spaces in iterative optimization, which is also an important benefit from a numerical and algorithmic standpoint. To support our claims, this paper has also provided experimental evidence regarding the practical advantages stemming from the use of MRPs in small as well as large-scale iterative optimization in classic problems in 3D computer vision.

Further advantages of MRPs include the flexibility in constructing smooth quaternion curves with minimal distortion in more intuitive ways. Specifically, we presented a novel general strategy for designing quaternion splines in the hyperplane by interpolating not only the key points, but also the derivatives of the spherical curve while working on its projection in the hyperplane. This yields smooth rational interpolants with minimal perspective distortion that are very competitive with popular algorithms such as spherical quadrangle interpolation (SQUAD).

Concluding, we briefly summarize the benefits of MRPs and stereographic projection as an orientation parameterizing scheme. It is a multi-purpose tool with convenient properties that allows for less complicated solutions in otherwise difficult, nonlinear or even intractable problems and offers efficiency up-to and beyond the standards of existing solutions as well as simplicity of design and implementation. C++ and Matlab code implementing most formulas in the paper is available in the following repository: <https://github.com/terzakig/Quaternion>.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A: Rotation Matrix Jacobian Tensor with Respect to MRPs

The derivatives of the elements of a rotation matrix $R = [r_{ij}]$ in terms of MRPs are simple quadratic expressions of the quaternion components, ρ and $v = [v_1 \ v_2 \ v_3]^T$ and they are obtained via the chain rule, using Eqs. 8 and 34. Since the Jacobian is a $3 \times 3 \times 3$ tensor, it is more convenient to give the gradient of each element separately:

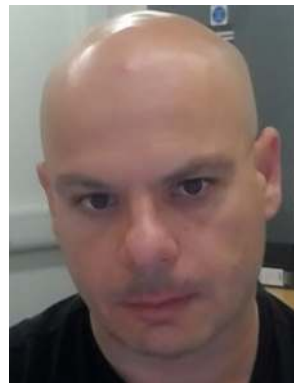
$$\begin{aligned} \frac{\partial r_{11}}{\partial \psi} &= 2 \begin{bmatrix} v_1 (-v_1^2 + v_2^2 + v_3^2 + 1 - \rho^2) \\ v_2 (-v_1^2 + v_2^2 + v_3^2 - (1 + \rho)^2) \\ v_3 (-v_1^2 + v_2^2 + v_3^2 - (1 + \rho)^2) \end{bmatrix}^T \\ \frac{\partial r_{12}}{\partial \psi} &= 2 \begin{bmatrix} v_2 (-2v_1^2 + \rho + 1) + v_1 v_3 (2\rho + 1) \\ v_1 (-2v_2^2 + \rho + 1) + v_2 v_3 (2\rho + 1) \\ -\rho (-2v_3^2 + \rho + 1) + v_3 (v_3 - 2v_1 v_2) \end{bmatrix}^T \\ \frac{\partial r_{13}}{\partial \psi} &= 2 \begin{bmatrix} v_3 (-2v_1^2 + \rho + 1) - v_1 v_2 (2\rho + 1) \\ \rho (-2v_2^2 + \rho + 1) - v_2 (v_2 + 2v_1 v_3) \\ v_1 (-2v_3^2 + \rho + 1) - v_2 v_3 (2\rho + 1) \end{bmatrix}^T \\ \frac{\partial r_{21}}{\partial \psi} &= 2 \begin{bmatrix} v_2 (-2v_1^2 + \rho + 1) - v_1 v_3 (2\rho + 1) \\ v_1 (-2v_2^2 + \rho + 1) - v_2 v_3 (2\rho + 1) \\ \rho (-2v_3^2 + \rho + 1) - v_3 (v_3 + 2v_1 v_2) \end{bmatrix}^T \\ \frac{\partial r_{22}}{\partial \psi} &= 2 \begin{bmatrix} v_1 (-v_2^2 + v_1^2 + v_3^2 - (\rho + 1)^2) \\ v_2 (-v_2^2 + v_1^2 + v_3^2 + 1 - \rho^2) \\ v_3 (-v_2^2 + v_1^2 + v_3^2 - (\rho + 1)^2) \end{bmatrix}^T \\ \frac{\partial r_{23}}{\partial \psi} &= 2 \begin{bmatrix} -\rho (-2v_1^2 + \rho + 1) + v_1 (v_1 - 2v_2 v_3) \\ v_3 (-2v_2^2 + \rho + 1) + v_1 v_2 (2\rho + 1) \\ v_2 (-2v_3^2 + \rho + 1) + v_1 v_3 (2\rho + 1) \end{bmatrix}^T \\ \frac{\partial r_{31}}{\partial \psi} &= 2 \begin{bmatrix} v_3 (-2v_1^2 + \rho + 1) + v_1 v_2 (2\rho + 1) \\ -\rho (-2v_2^2 + \rho + 1) + v_2 (v_2 - 2v_1 v_3) \\ v_1 (-2v_3^2 + \rho + 1) + v_2 v_3 (2\rho + 1) \end{bmatrix}^T \\ \frac{\partial r_{32}}{\partial \psi} &= 2 \begin{bmatrix} \rho (-2v_1^2 + \rho + 1) - v_1 (v_1 + 2v_2 v_3) \\ v_3 (-2v_2^2 + \rho + 1) - v_1 v_2 (2\rho + 1) \\ v_2 (-2v_3^2 + \rho + 1) - v_1 v_3 (2\rho + 1) \end{bmatrix}^T \end{aligned}$$

$$\frac{\partial r_{33}}{\partial \psi} = 2 \begin{bmatrix} v_1 (-v_3^2 + v_1^2 + v_2^2 - (\rho + 1)^2) \\ v_2 (-v_3^2 + v_1^2 + v_2^2 - (\rho + 1)^2) \\ v_3 (-v_3^2 + v_1^2 + v_2^2 + 1 - \rho^2) \end{bmatrix}^T$$

References

- Absil, P.A., Mahony, R., Sepulchre, R.: *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton (2009)
- Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: *European Conference on Computer Vision (ECCV)*, pp. 29–42. Springer, Berlin Heidelberg (2010)
- Barr, A.H., Currin, B., Gabriel, S., Hughes, J.F.: Smooth interpolation of orientations with angular velocity constraints using quaternions. *SIGGRAPH Comput. Gr.* **26**(2), 313–320 (1992)
- Bauchau, O.A., Trainelli, L.: The vectorial parameterization of rotation. *Nonlinear Dyn.* **32**(1), 71–92 (2003)
- Boumal, N.: Interpolation and regression of rotation matrices. In: Nielsen, F., Barbaresco, F. (eds.) *Geometric Science of Information*, pp. 345–352. Springer, Berlin, Heidelberg (2013)
- Briaies, J., Gonzalez-Jimenez, J.: Convex global 3D registration with Lagrangian duality. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)
- Carlone, L., Rosen, D.M., Calafiore, G., Leonard, J.J., Dellaert, F.: Lagrangian duality in 3D SLAM: verification techniques and optimal solutions. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 125–132. IEEE (2015)
- Casler Jr, R.J., Penkar, R.C.: Multiaxis robot control having improved continuous path operation. US Patent 4,772,831 (1988)
- Catmull, E., Rom, R.: A class of local interpolating splines. *Comput. Aided Geom. Des.* **74**, 317–326 (1974)
- Chatterjee, A., Madhav Govindu, V.: Efficient and robust large-scale rotation averaging. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 521–528 (2013)
- Corke, P.: *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, Berlin (2013)
- Dai, Y., Trunpf, J., Li, H., Barnes, N., Hartley, R.: Rotation averaging with application to camera-rig calibration. In: *Asian Conference on Computer Vision*, pp. 335–346. Springer (2009)
- Dam, E.B., Koch, M., Lillholm, M.: Quaternions, interpolation and animation. Tech. Rep. DIKU-TR-98/5, Department of Computer Science. University of Copenhagen, Copenhagen (1998)
- Dash, A.K., Chen, I.M., Yeo, S.H., Yang, G.: Workspace generation and planning singularity-free path for parallel manipulators. *Mech. Mach. Theory* **40**(7), 776–805 (2005)
- Davenport, P.B.: A vector approach to the algebra of rotations with applications, NASA technical note, vol. 4696. National Aeronautics and Space Administration (1968)
- Diebel, J.: Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **58**(15–16), 1–35 (2006)
- Dietz, R., Hoschek, J., Jüttler, B.: An algebraic approach to curves and surfaces on the sphere and on other quadrics. *Comput. Aided Geom. Des.* **10**(3–4), 211–229 (1993)
- Drummond, T., Cipolla, R.: Application of Lie algebras to visual servoing. *Int. J. Comput. Vis.* **37**(1), 21–41 (2000)
- Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 932–946 (2002)
- Duff, T.: *Splines in animation and modeling*. State of the Art in Image Synthesis (SIGGRAPH'86 course notes no. 15, Dallas, TX) (1986)
- Farouki, R.T., Al-Kandari, M., Sakkalis, T.: Structural invariance of spatial Pythagorean hodographs. *Comput. Aided Geom. Des.* **19**(6), 395–407 (2002)
- Feng-yun, L., Tian-sheng, L.: Development of a robot system for complex surfaces polishing based on CL data. *Int. J. Adv. Manuf. Technol.* **26**(9–10), 1132–1137 (2005)
- Gallego, G., Yezzi, A.: A compact formula for the derivative of a 3-D rotation in exponential coordinates. *J. Math. Imag. Vis.* **51**(3), 378–384 (2015)
- Hartley, R., Aftab, K., Trunpf, J.: L1 rotation averaging using the Weiszfeld algorithm. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3041–3048. IEEE (2011)
- Hartley, R., Trunpf, J., Dai, Y., Li, H.: Rotation averaging. *Int. J. Comput. Vis.* **103**(3), 267–305 (2013)
- Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN: 0521540518
- Heise, R., MacDonald, B.A.: Quaternions and motion interpolation: A tutorial. In: *New Advances in Computer Graphics*, pp. 229–243. Springer (1989)
- Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. *JOSA A* **4**(4), 629–642 (1987)
- Horn, B.K., Hilden, H.M., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A* **5**(7), 1127–1135 (1988)
- Hughes, J.F., van Dam, A., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., Akeley, K.: *Computer Graphics: Principles and Practice*. The Systems Programming Series. Addison-Wesley, Boston (2014)
- Johnstone, J.K., Williams, J.P.: Rational control of orientation for animation. In: *Graphics Interface*, pp. 179–179. Canadian Information Processing Society, (1995)
- Kim, M.J., Kim, M.S., Shin, S.Y.: A general construction scheme for unit quaternion curves with simple high order derivatives. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 369–376. ACM (1995)
- Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 225–234. IEEE (2007)
- Levenberg, K.: A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **2**(2), 164–168 (1944)
- Lourakis, M.: Sparse non-linear least squares optimization for geometric vision. *European Conference on Computer Vision (ECCV)* **2**, 43–56 (2010)
- Lourakis, M.: An efficient solution to absolute orientation. In: *International Conference on Pattern Recognition (ICPR)*, pp. 3816–3819 (2016)
- Lourakis, M., Argyros, A.: SBA: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Software* **36**(1), 1–30 (2009). www.ics.forth.gr/~lourakis/sba
- Lourakis, M., Zabulis, X.: Model-based pose estimation for rigid objects. In: *International Conference on Computer Vision Systems (ICVS)*, pp. 83–92. Springer (2013). www.ics.forth.gr/~lourakis/posest
- Lui, V., Drummond, T.: An iterative 5-pt algorithm for fast and robust essential matrix estimation. In: *BMVC* (2013)
- Markley, F.L., Crassidis, J.L.: *Fundamentals of Spacecraft Attitude Determination and Control*, Space Technology Library. Springer, Berlin (2014)
- Markley, F.L., Mortari, D.: Quaternion attitude estimation using vector observations. *J. Astronaut. Sci.* **48**(2), 359–380 (2000)
- Marquardt, D.W.: An algorithm for least-squares estimation of non-linear parameters. *J. Soc. Ind. Appl. Math.* **11**(2), 431–441 (1963)
- Olsson, C., Eriksson, A.: Solving quadratically constrained geometrical problems using lagrangian duality. In: *International Conference on Pattern Recognition (ICPR)*, pp. 1–5. IEEE (2008)

44. Pletinckx, D.: Quaternion calculus as a basic tool in computer graphics. *Vis. Comput.* **5**(1–2), 2–13 (1989)
45. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. *Int. J. Comput. Vis.* **59**(3), 207–232 (2004)
46. Ramamoorthi, R., Ball, C., Barr, A.H.: Dynamic splines with constraints for animation. Tech. Rep. 03. California Institute of Technology, Pasadena, CA (1997)
47. Roberts, K.S., Bishop, G., Ganapathy, S.K.: Smooth interpolation of rotational motions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 724–729 (1988)
48. Rodrigues, O.: Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J. de Mathématiques Pures et Appliquées* **5**, 380–440 (1840)
49. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and adverbs: multi-dimensional motion interpolation. *IEEE Comput. Gr. Appl.* **18**(5), 32–40 (1998)
50. Schaub, H., Junkins, J.L.: Stereographic orientation parameters for attitude dynamics: a generalization of the Rodrigues parameters. *J. Astronaut. Sci.* **44**(1), 1–19 (1996)
51. Schaub, H., Junkins, J.L.: *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston (2003)
52. Schaub, H., Tsiotras, P., Junkins, J.L.: Principal rotation representations of proper $n \times n$ orthogonal matrices. *Int. J. Eng. Sci.* **33**(15), 2277–2295 (1995)
53. Schulz, V., Bock, H., Longman, R.: Shortest paths for satellite mounted robot manipulators. In: *Computational optimal control*, pp. 357–366. Springer (1994)
54. Selig, J.M.: Lie groups and lie algebras in robotics. In: *Computational Noncommutative Algebra and Applications*, pp. 101–125. Springer (2004)
55. Shen, Y., Huper, K., Leite, F.S.: Smooth interpolation of orientation by rolling and wrapping for robot motion planning. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 113–118. IEEE (2006)
56. Shoemake, K.: Animating rotation with quaternion curves. *SIGGRAPH Comput. Gr.* **19**(3), 245–254 (1985)
57. Shoemake, K.: Quaternion calculus and fast animation, computer animation: 3-D motion specification and control. *SIGGRAPH '87 tutorial* (1987)
58. Shuster, M.D.: A survey of attitude representations. *J. Astronaut. Sci.* **41**(4), 439–517 (1993)
59. Szeliski, R.: *Computer Vision: Algorithms and Applications*. Springer, New York (2010)
60. Taylor, C.J., Kriegman, D.J.: Minimization on the Lie group $SO(3)$ and related manifolds. Tech. Rep. 9405. Yale University, New Haven, CT (1994)
61. Terzakis, G., Culverhouse, P., Bugmann, G., Sharma, S., Sutton, R.: A recipe on the parameterization of rotation matrices for non-linear optimization using quaternions. Tech. Rep. 004, School of Marine Science and Engineering, Plymouth University, (2012)
62. Tsiotras, P., Junkins, J.L., Schaub, H.: Higher-order Cayley transforms with applications to attitude representations. *J. Guid. Control Dyn.* **20**(3), 528–534 (1997)
63. Tsiotras, P., Longuski, J.M.: A new parameterization of the attitude kinematics. *J. Astronaut. Sci.* **43**(3), 243–262 (1995)
64. Vicci, L.: Quaternions and rotations in 3-space: The algebra and its geometric interpretation. Tech. Rep. 01-014, Dept. of Computer Science, UNC at Chapel Hill (2001)
65. Wahba, G.: A least squares estimate of satellite attitude. *SIAM Rev.* **7**(3), 409–409 (1965)
66. Watt, A.: *3D computer graphics*. Addison-Wesley, Boston (1993)
67. Watt, A.: *3D Computer Graphics*, 3rd edn. Addison-Wesley, Boston (2000)
68. Wiener, T.F.: Theoretical analysis of gimballess inertial reference equipment using delta-modulated instruments. Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Mass, USA (1962)



mapping in natural environments.



matching, multiple and single view geometry, camera(s) self-calibration, 3D reconstruction, 3D shape analysis and matching, and continuous optimization.

George Terzakis is a SLAM engineer at Dynium Robot and a visiting research associate at the University of Portsmouth. He is a computer scientist-turned-roboticist with research interests focused on 3D computer vision, augmented reality, mobile robotics and graphics. His academic degrees include a 4-year BSc in computer science, a 2-year MSc in Electronics and Computer Technology, a 1-year MSc in Robotics and a PhD in vision based localization and

Manolis Lourakis is a principal researcher at the Computational Vision and Robotics Lab of the Institute of Computer Science, FORTH in Heraklion, Greece. In 1999, he received the Ph.D. degree in Computer Science from the University of Crete, Greece and has been with FORTH since 2002, having worked on several R&D projects. His research interests focus on computer vision and span topics related to robotics, motion tracking and analysis, registration and



Djamel Ait-Boudaoud joined the University in 2010 and he is currently Dean of the Faculty of Technology. He has held senior roles in the UK and internationally, and before joining the University he was the head of the School of Computing, Engineering and Physical Sciences at the University of Central Lancashire for close to 10 years. He gained a PhD in 1991 from the Department of Electrical and Electronic Engineering at University of Nottingham, UK. He

is a Chartered Engineer (CEng) and a fellow of the Institution of Engineering and Technology (FIET). His research interests are predominantly focused on the problems of optimization with applications in 3D computer vision, video standards (H264) and solving combinatorial (ordered-sequence) problems using evolutionary algorithms. He is currently a committee member of the Engineering Professors' Council (EPC) and serves on the readers' panel for the Queen's Anniversary Prize for Higher and Further Education, UK.