Motion reconstruction for optical tomography of trapped objects

Peter Elbau1, Monika Ritsch-Marte2, Otmar Scherzer1,3 and Denise Schmutz1,4

1 Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria
2 Division for Biomedical Physics, Medical University of Innsbruck, Müllerstraße 44, A-6020 Innsbruck, Austria
3 Johann Radon Institute for Computational and Applied Mathematics (RICAM), Altenbergerstraße 69, A-4040 Linz, Austria

E-mail: peter.elbau@univie.ac.at, monika.ritsch-marte@i-med.ac.at, otmar.scherzer@univie.ac.at and denise.schmutz@univie.ac.at

Received 27 June 2019, revised 21 November 2019
Accepted for publication 6 January 2020
Published 4 March 2020

Abstract
Optical and acoustical trapping has been established as a tool for holding and moving microscopic particles suspended in a liquid in a contact-free and non-invasive manner. Opposed to standard microscopic imaging, where the probe is fixated, this technique allows imaging in a more natural environment. This paper provides a method for estimating the movement of a transparent particle which is maneuvered by tweezers (assuming that the inner structure of the probe is not subject to local movements) by making use of the assumption of a smooth movement in time. The mathematical formulation of the motion estimation leads to an infinitesimal version of the common line technique used in cryogenic electron microscopy single particle imaging to estimate the orientations of the particles in the probe.

Keywords: optical trapping, motion estimation, x-ray-transform, optical tomography, Cryo-EM, acoustical trapping, common lines

(Some figures may appear in colour only in the online journal)

Original content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Author to whom any correspondence should be addressed.
1. Introduction

In the past decades optical trapping has established itself as a versatile, and easy-to-implement tool for holding and moving microscopic (µm-sized) particles suspended in a liquid in a contact-free and non-invasive manner [8]. The optical forces needed to balance the weight scale with the particle volume. Thus, for increasing particle size heating of the sample due to absorption of the trapping light leads to a problem, except for very transparent specimens. To hold heavier samples (for example cell-clusters, entire micro-organisms, or so-called ‘organoids’, mini-organs grown in the petri-dish) other trapping modalities, such as acoustic forces, have to be added as auxiliary trapping forces [14].

In this paper, we study the problem of estimating the movement of trapped particles, as a preprocessing step for 3D tomographic imaging. Conveniently, the required tomographic projection images from different view-points can be taken by rotating the trapped specimen directly by means of the acoustic or optical trapping forces. Here, we will assume to be in a setting where the light propagation can be sufficiently well modeled by straight rays which only undergo spatially varying attenuation; as in the model of x-ray tomography. In this case the optical image resembles a projection image. This is, for instance, fulfilled in low numerical aperture imaging of biological samples with sufficient amplitude contrast and with large structures compared to the scale of the optical wavelength. For high-resolution imaging with a high numerical aperture objective, or for samples with small structures diffracting the light beams significantly, deviations from geometrical optics are to be expected. Then, optical projection tomography based on the Fourier slice theorem will have to be replaced by optical diffraction tomography based on the Fourier diffraction theorem, as derived in [10].

In contrast to classical computerized tomography, however, the motion of the particle in the trap is irregular: since the particle is not completely immobilized in the trap and since the locally acting forces are typically inhomogeneous during an entire revolution, it undergoes a smooth, but complicated motion described by an affine transformation with time-dependent—and unknown—parameters, the momentary translation and angular velocity vectors. The focus of this work is to provide, as a first step of the reconstruction, a recovery of the motion of the particle without knowledge of the inner structure. As soon as the motion is determined, the attenuation projection data can be properly aligned to give the three-dimensional x-ray transform (at least for those directions which were attained during the motion) and from this, a standard reconstruction formula could be applied to recover the inner structure of the object, see, for example, [7].

A main ingredient for our reconstruction of the motion is the common line method known for the determination of the relative orientations of different projection images as it is used, for example, in cryogenic single-particle electron microscopy (cryo-EM), see [5, 6]. An infinitesimal version of this common line technique, allows us (after correcting for a potential translation by tracking the center of the images, as explained in section 3.1) to find at every time step the local angular velocity of the motion from an infinitesimally short observation time, see section 3.2. As a numerical verification of the algorithm, we provide in section 4 reconstructions of simulated data.

2. Mathematical model

We consider a bounded object in $\mathbb{R}^3$, characterized by an attenuation coefficient $u \in C_{c, \nu}(\mathbb{R}^3, \mathbb{R})$. 

---

**P Elbau et al**

*Inverse Problems 36 (2020) 044004*
Definition 2.1. We denote by
\[ C_{c,+}(\mathbb{R}^3; \mathbb{R}) = \{ u \in C(\mathbb{R}^3; \mathbb{R}) | \text{supp}(u) \neq \emptyset \text{ is compact, } u \geq 0 \} \]
the space of admissible attenuation coefficients. For \( u \in C_{c,+}(\mathbb{R}^3; \mathbb{R}) \) we define its center
\[ C_3 := \frac{1}{\int_{\mathbb{R}^3} u(x)dx} \int_{\mathbb{R}^3} xu(x)dx \in \mathbb{R}^3 \text{ with } x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \] (2.1)

In this paper we make the assumption that the object does not undergo a deformation during its motion, that is:

Assumption 2.2. The attenuation coefficient \( u \) is exposed to a rigid motion consisting of a rotation \( R \in C(\mathbb{R}; \text{SO}(3)) \) and a translation \( T \in C(\mathbb{R}; \mathbb{R}^3) \),
\[ A(t, x) = C_3 + R(t)(x - C_3 + T(t)). \] (2.2)

During such an induced motion, the object is illuminated from the direction \( e_3 \) (denoting by \( e_1, e_2, e_3 \) the standard basis vectors in \( \mathbb{R}^3 \)) with a uniform intensity, see figure 1.

In the experiments, the induced motion of the object turned out to be sufficiently smooth on every relevant time scale. As a first approximation for the problem, we will assume to be in a regime where the light propagation can be sufficiently well described by geometrical optics so that we can consider the light to move along straight lines suffering only from attenuation. Then, the measurements can be modeled in the form of attenuation projection mappings.

Definition 2.3. The attenuation mapping \( J \) maps a rigid motion (described by a translation \( t \mapsto T(t) \) and a rotation \( t \mapsto R(t) \)) applied to an object of interest (described by the attenuation function \( u : \mathbb{R}^3 \rightarrow \mathbb{R} \)) onto the attenuation projection image. That is the attenuation mapping of \( u \) is given by
\[ (T, R) \mapsto J[T, R](t, x_1, x_2) = \int_{-\infty}^{\infty} u \left( C_3 + R(t) \left( \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - C_3 + T(t) \right) \right) dx_3, \ t \in \mathbb{R}, (x_1, x_2) \in \mathbb{R}^2. \] (2.3)

We will do the reconstruction of the motion in Fourier space, using the convention
\[ \mathcal{F}_n[f](k) = (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(x)e^{-i(k,x)}dx, \ f \in L^1(\mathbb{R}^3), \]
for the \( n \)-dimensional Fourier transform. And it is convenient to use the orthogonal projection
\[ P : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \]
\[ x \mapsto \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \]
on the \( x_1x_2 \)-plane and its adjoint \( P^T : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \), given by \( P^Tk = (k, 0)^T \), to shorten the notation.

In Fourier space, the attenuation mapping \( J \) satisfies equation (2.4), below, which is an identity similar to the Fourier projection-slice theorem, see for example [3] and [11, p 11], that is used quite heavily for orientation reconstruction in cryo-EM and x-ray tomography. It also shows that if the motion \( (T, R) \) is known, then the Fourier transform of \( J \) directly allows us to
recover the Fourier transform of the attenuation coefficient \( u \) for the corresponding directions, thus providing us a way to fully reconstruct the attenuation coefficient.

**Lemma 2.4.** Let \( u \in C_{c,+}(\mathbb{R}^3; \mathbb{R}) \) and \( \mathcal{J}[R, T] \) be the attenuation mapping of a rigid body motion \((R, T)\), which the object of interest \( u \) gets exposed to. Then, the following identity holds:

\[
\mathcal{F}_2[\mathcal{J}[T, R]](t, k) = \sqrt{2\pi} \mathcal{F}_3[u(R(t)P^T k)] e^{i\langle P^T k, C_3 \rangle} e^{i\langle k, P(T(t) - C_3) \rangle}, \quad t \in \mathbb{R}, \quad k \in \mathbb{R}^2.
\]  

(2.4)

Here, \( \mathcal{F}_2 \) denotes the two-dimensional Fourier transform with respect to the coordinates \((x_1, x_2)\).

**Proof.** We denote by

\[
\hat{\mathcal{J}} := \mathcal{F}_2[\mathcal{J}[T, R]]
\]

the Fourier transform of the attenuation mapping of \( u \) exposed to a rigid transformation. By the definition of \( \mathcal{J} \), see equation (2.3), we then have that

\[
\hat{\mathcal{J}}(t, k) = \frac{1}{2\pi} \int_{\mathbb{R}^2} \mathcal{J}[T, R](t, x_1, x_2) e^{-i k_1 x_1 - i k_2 x_2} d(x_1, x_2)
\]

\[
= \frac{1}{2\pi} \int_{\mathbb{R}^3} u(C_3 + R(t)(x - C_3 + T(t))) e^{-i(p^T k_1)} dx.
\]

Substituting \( y = C_3 + R(t)(x - C_3 + T(t)) \) in the integral, we obtain

\[
\hat{\mathcal{J}}(t, k) = \frac{1}{2\pi} \int_{\mathbb{R}^3} u(y) e^{-i(p^T k_1)(y-C_3)} dy,
\]

which, upon inserting the definition of \( \mathcal{F}_3[u] \), is exactly equation (2.4). \( \square \)

This means that the Fourier transform of the two-dimensional attenuation mapping \( \hat{\mathcal{J}} \) of a rigid motion \((T, R)\) of an object of interest \( u \) coincides, up to a factor depending on the motion \( A \), with the values of the three-dimensional Fourier transform of \( u \) evaluated on the central slice spanned by the first two columns of \( R(t) \).
3. Motion estimation

In this section we present a method for the reconstruction of the horizontal components of the time dependent translation \( t \mapsto T(t) \) and the rotation \( t \mapsto R(t) \) from the collected data \( \mathcal{J}[T, R] \), as defined in equation (2.3). For this purpose we

(i) show in section 3.1 how to determine from the centers of the attenuation mappings \( \mathcal{J}[T, R] \) the two components \( T_1 \) and \( T_2 \) of the translation;

(ii) define accordingly in equation (3.7) the reduced attenuation mapping \( \tilde{\mathcal{J}} \) which does not depend on the translation \( T \) anymore; and

(iii) use in section 3.2 the first and third order terms of our formulation of the common line property, see lemma 3.6, to determine the rotation \( R \).

Instead of recovering the rotation matrix function \( t \mapsto R(t) \) directly, we rather retrieve the corresponding angular velocity function \( t \mapsto \omega(t) \), which is defined as follows.

**Definition 3.1.** Let \( R \in C^4(\mathbb{R}; SO(3)) \). Then the angular velocity \( \omega \in C(\mathbb{R}; \mathbb{R}^3) \) corresponding to \( R \) is defined via the relation

\[
R(t)^T R'(t)x = \omega(t) \times x \quad \text{for all } t \in \mathbb{R}, \ x \in \mathbb{R}^3,
\]

where \( R'(t) \) denotes the derivative with respect to the time \( t \).

Conveniently, we represent \( \omega \) in cylindrical coordinates with cylindrical axis \( e_3 \) and denote by \( \alpha \in C(\mathbb{R}; \mathbb{R}^+ \mathbb{R}) \) the cylindrical radius of \( \omega \), and by \( \nu \in C(\mathbb{R}; \mathbb{S}^1) \) the cylindrical components of \( \omega \), and by \( \varphi \in C(\mathbb{R}; \mathbb{R}) \) the cylindrical angle of \( \omega \), such that

\[
\omega(t) = \begin{pmatrix} \alpha(t) \nu_1(t) \\ \omega_3(t) \end{pmatrix} = \begin{pmatrix} (\alpha(t) \cos(\varphi(t))) \\ (\alpha(t) \sin(\varphi(t))) \end{pmatrix}.
\]

Moreover, we set \( \nu^\perp(t) = (-\nu_2(t), \nu_1(t))^T \).

The reconstruction algorithm thus follows the steps shown in figure 2.

3.1. Reconstruction of the translation

Since the attenuation operator integrates along the direction \( e_3 \), we have an invariance of the attenuation projection data with respect to translations of the object in this direction.

**Lemma 3.2.** Let \( u \in C_{c,+}(\mathbb{R}^3; \mathbb{R}) \) and \( \mathcal{J} \) be the attenuation mapping of a rigid motion of \( u \). Then,

\[
\mathcal{J}[T, R] = \mathcal{J}[T + \rho e_3, R]
\]

for every \( T \in C(\mathbb{R}; \mathbb{R}) \), \( \rho \in C(\mathbb{R}; \mathbb{R}) \), and \( R \in C(\mathbb{R}; SO(3)) \).

**Proof.** Substituting \( \tilde{x}_3 = x_3 + \rho(t) \) in the definition of \( \mathcal{J}[T + \rho e_3, R] \), see equation (2.3), we find that

\[
\mathcal{J}[T + \rho e_3, R](t, x_1, x_2) = \int_{-\infty}^{\infty} u \left( C_3 + R(t) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \rho(t) e_3 - C_3 + T(t) \right) \, dx_3
\]

\[
= \int_{-\infty}^{\infty} u \left( C_3 + R(t) \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix} - C_3 + T(t) \right) \, d\tilde{x}_3 = \mathcal{J}[T, R](t, x_1, x_2).
\]

\[\square\]
It is therefore impossible to reconstruct the third component of the translation $T$ of the object of interest (that is the attenuation function), and thus we aim to recover only the two components $T_1$ and $T_2$. Now, the orthogonal projection of the center of the translated and rotated object $u$ is equal to the center of the corresponding attenuation mapping, which can be directly computed from the projection data.

**Proposition 3.3.** Let $u \in C_c(\mathbb{R}^3; \mathbb{R})$ and $\mathcal{J}$ be the attenuation mapping of a rigid motion of the object of interest $u$ is exposed to. Moreover, let $C_3 \in \mathbb{R}^3$ be the center of $u$ defined in equation (2.1) and $C_3 : \mathbb{R} \to \mathbb{R}^3$ be the two-dimensional center of the attenuation mapping of $\mathcal{J}[T, R]$ for some translation $T \in C(\mathbb{R}; \mathbb{R}^3)$ and some rotation $R \in C(\mathbb{R}; SO(3))$, that is,

$$C_3(t) := \frac{1}{\int_{\mathbb{R}^2} \mathcal{J}[T, R](t, x_1, x_2)dx_1 dx_2} \int_{\mathbb{R}^2} x_1 \mathcal{J}[T, R](t, x_1, x_2)dx_1 dx_2, \quad t \in \mathbb{R}. \quad (3.3)$$

Then, we have

$$P(C_3 - T(t)) = C_3(t) \text{ for every } t \in \mathbb{R}. \quad (3.4)$$

**Proof.** To calculate the center of $\mathcal{J}[T, R]$, we first remark that the denominator in equation (3.3) is simply the total attenuation of the object, which can be seen explicitly by inserting the definition of $\mathcal{J}$, see equation (2.3), and by substituting $y = C_3 + R(t)(x - C_3 + T(t))$:

$$\int_{\mathbb{R}^2} \mathcal{J}[T, R](t, x_1, x_2)dx_1 dx_2 = \int_{\mathbb{R}^3} u(C_3 + R(t)(x - C_3 + T(t)))dx = \int_{\mathbb{R}^3} u(y)dy. \quad (3.5)$$

In the same way, we find

$$\int_{\mathbb{R}^2} x_1 \mathcal{J}[T, R](t, x_1, x_2)dx_1 dx_2 = \int_{\mathbb{R}^3} Pxu(C_3 + R(t)(x - C_3 + T(t)))dx$$

$$= \int_{\mathbb{R}^3} P[R(t)^T(y - C_3) + C_3 - T(t)]u(y)dy$$

$$= PR(t)^T \left( \int_{\mathbb{R}^3} yu(y)dy - C_3 \int_{\mathbb{R}^3} u(y)dy \right) + P(C_3 - T(t)) \int_{\mathbb{R}^3} u(y)dy.$$
which is just equation (3.4).

From equation (3.4), we immediately get that
\[ P(T(t) - T(0)) = C_2(0) - C_2(t) \text{ for every } t \in \mathbb{R}, \]
which determines the components \( T_1 \) and \( T_2 \) up to a global translation \( P(T(0)) \) (which is due to the fact that the placement of the object \( u \) is not fixed in any way).

3.2. Reconstruction of the rotation

In a second step we aim at recovering the cylindrical parameters \( t \mapsto \varphi(t) \) (\( t \mapsto \varphi(t) \), respectively), \( t \mapsto \alpha(t) \) and \( t \mapsto \omega_3(t) \) corresponding to the rotation function \( t \mapsto R(t) \) from the reduced attenuation mapping \( \tilde{J} \), which is the Fourier transform of \( J \) after translation correction with equation (3.4):

**Definition 3.4.** We define the reduced attenuation mapping corresponding to an admissible attenuation coefficient \( u \) under a motion \((T, R)\) as
\[
\tilde{J} : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R},
(t, k) \mapsto \mathcal{F}_2[J[T, R]](t, k) e^{-i(k \cdot C_3(t))}.
\]

**Lemma 3.5.** The reduced attenuation mapping of an admissible attenuation coefficient \( u \) under a motion \((T, R)\) is independent of the translation \( t \mapsto T(t) \).

**Proof.** Multiplying equation (2.4) with \( e^{-i(k \cdot C_3(t))} \) and taking into account the relation between \( C_2 \) and \( C_3 \) from equation (3.4) we get the expression
\[
\tilde{J}(t, k) = \sqrt{2\pi} \mathcal{F}_3[u](R(t)P^T k) e^{i(R(t)P^T k \cdot C_3)}
\]
for the reduced attenuation mapping, which is seen to be independent of the translation \( T \).

The reduced attenuation mapping possesses the following symmetry property, which is the common line property known from methods in cryo-EM, see, for example, [5, 6].

**Lemma 3.6.** Let \( u \in C_{c,+}(\mathbb{R}^3; \mathbb{R}) \) and \( R \in C(\mathbb{R}; SO(3)) \). Then, the reduced attenuation mapping \( \tilde{J} \) of \( u \) under the rotation \( R \) fulfills the relation
\[
\tilde{J} \left( s, \frac{\lambda}{s - t} P(e_3 \times (R(s)^T R(t)e_3)) \right) = \tilde{J} \left( t, \frac{\lambda}{s - t} P(e_3 \times (R(t)^T R(s)e_3)) \right) \text{ for all } \lambda \in \mathbb{R}
\]
for all values \( s, t \in \mathbb{R} \) with \( s \neq t \).

**Proof.** We remark that we have for arbitrary \( s, t \in \mathbb{R} \) the relation
\[
(R(t)e_3) \times (R(s)e_3) = R(t)(e_3 \times (R(t)^T R(s)e_3)) = R(t)P^T P(e_3 \times (R(t)^T R(s)e_3)).
\]
Choosing for \( s \neq t \) the vector
\[
k = \frac{\lambda}{s - t} P(e_3 \times (R(t)^T R(s)e_3))
\]
with some arbitrary $\lambda \in \mathbb{R}$ in equation (3.8), we find that

$$
\tilde{J} \left( t, \frac{\lambda}{s - t}P(e_3 \times (R(t)^T R(s)e_3)) \right) = \sqrt{2\pi J_3} \left( \frac{\lambda}{s - t} (R(t)e_3) \times (R(s)e_3) \right) e^{i \frac{\lambda}{s - t} \langle (R(t)e_3) \times (R(s)e_3), C_3 \rangle}.
$$

(3.10)

Now, the right hand side of equation (3.10) is because of the skew-symmetry of the cross product invariant with respect to interchanging $s$ and $t$, and we therefore get equation (3.9).

**Remark 1.** Since $\lim_{t \to s} P(e_3 \times (R(s)^T R(t)e_3)) = 0$, we chose the scaling $\frac{\lambda}{s - t}$ for the free scalar parameter in equation (3.9) so that the argument $\frac{\lambda}{s - t}P(e_3 \times (R(s)^T R(t)e_3))$ converges in the later considered limit $t \to s$ in general (that is, for $\alpha(s) \neq 0$) to a finite value.

Based on this identity, the relative orientation of the object at different times can be determined, which is also extensively used in cryo-EM, see, for example, [4, 13, 15]. In the following, we want to show a way to explore the smooth time dependence to calculate from this relation locally at every time the change in movement (that is, calculating the angular velocity $\omega$) by taking time derivatives of equation (3.9), which is an information not available in cryo-EM.

### 3.2.1. Reconstruction of the cylindrical and third components of the angular velocity.

We show below that by differentiation of equation (3.9) with respect to $t$ we can recover the cylindrical components $v$ and the third component $\omega_3$ of the angular velocity $\omega$ as defined in equations (3.2) and (3.1), respectively. In this section we use the tensor derivative notation as summarized in the appendix (see section A.1 in the appendix).

**Proposition 3.7.** Let $u \in C_{c,+}(\mathbb{R}^3; \mathbb{R})$ be an admissible attenuation coefficient and $R \in C^2(\mathbb{R}; SO(3))$ be a rotation with associated angular velocity $\omega \in C^1(\mathbb{R}; \mathbb{R}^3)$ defined by equation (3.1). Let further $\tilde{J}$ be the reduced attenuation map corresponding to $u$ under the rotation $R$ as defined in equation (3.7).

Then, we have for all $t \in \mathbb{R}$ satisfying $\alpha(t) \neq 0$ and all $\mu \in \mathbb{R}$ the relation

$$
\partial_t \tilde{J}(t, \mu \omega(t)) = \omega_3(t) D^1[\tilde{J}](t, \mu \omega(t))[\mu \omega^1(t)].
$$

(3.11)

**Proof.** We insert in equation (3.9) the first-order Taylor polynomials

$$
\frac{1}{t - s}P(e_3 \times (R(s)^T R(t)e_3)) = a_0(s) + a_1(s)(t - s) + o(t - s) \text{ and } \frac{1}{s - t}P(e_3 \times (R(t)^T R(s)e_3)) = b_0(s) + b_1(s)(t - s) + o(t - s)
$$

with the coefficients $a_j$ and $b_j$, $j = 0, 1$, being calculated in lemma A.1. Then, by differentiating equation (3.9) with respect to $t$ at the position $t = s$, we find for every $\lambda \in \mathbb{R}$ the relation

$$
D^1[\tilde{J}](s, \lambda a_0(s))[\lambda a_1(s)] = \partial_s \tilde{J}(s, \lambda b_0(s)) + D^1[\tilde{J}](s, \lambda b_0(s))[\lambda b_1(s)].
$$

Inserting the expressions for the coefficients $a_j$ and $b_j$, $j = 0, 1$, from equation (A.3), we end up with the identity


\[ \partial_s \tilde{\mathcal{J}}(s, \lambda \alpha(s) v(s)) = D_1 [\tilde{\mathcal{J}}(s, \lambda \alpha(s) v(s))] [\lambda \omega_3(s) \alpha(s) v^\perp(s)] \] for all \( \lambda, s \in \mathbb{R} \).  
\[ (3.12) \]

If we choose for an \( s \in \mathbb{R} \) with \( \alpha(s) \neq 0 \) the parameter \( \lambda = \frac{\mu}{\alpha(s)} \), this gives us equation (3.11).

Thus, the cylindrical components \( v(t) \) and the third component \( \omega_3(t) \) satisfy for every time \( t \in \mathbb{R} \) the equation (3.11) and we can therefore attempt to reconstruct these parameters by looking for a vector \( \tilde{\nu} \in \mathbb{S}^1 \) such that the function

\[
\Psi_E : \mu \mapsto \frac{\partial_s \tilde{\mathcal{J}}(t, \mu \tilde{\nu})}{D_1[\tilde{\mathcal{J}}(t, \tilde{\nu})][\mu \tilde{\nu}^\perp]} \tag{3.13}
\]

is finite and constant on the set of all values \( \mu \in \mathbb{R} \) for which numerator and denominator do not both vanish. Ideally, there would exist a unique vector \( \tilde{\nu} \) with this property, which would then necessarily give us the cylindrical components \( v(t) = \tilde{\nu} \) of the angular velocity and the constant value of the function in equation (3.13) would equal to the third component:

\( \omega_3(t) = \Psi_E(\mu) \) for every \( \mu \in \mathbb{R} \) where \( \Psi_E \) is well-defined.

Now, the uniqueness can at best only be up to a choice of sign, since with \( v(t) \) also \( -v(t) \) fulfills the property of proposition 3.7 (with the same value \( \omega_3(t) \)). In fact, this non-uniqueness is already present in equation (3.9), since it is not only invariant under global rotations \( R(t) \mapsto R_0 R(t) \) for some \( R_0 \in SO(3) \) (reflecting that the orientation of the attenuation coefficient can be arbitrarily chosen), but also under the orientation changing coordinate transform \( \Sigma = \text{diag}(1, 1, -1) \), which acts on the rotation via \( R(t) \mapsto \tilde{R}(t) = \Sigma^T R(t) \Sigma \), see figure 3 for an illustration, and for which the angular velocity \( \tilde{\omega} \) corresponding to the transformed rotation \( \tilde{R} \) is, because of

\[
\tilde{R}'(t)x = \Sigma^T R'(t) \Sigma x = \Sigma^T R(t)(\omega(t) \times (\Sigma x)) = -R(t)(\Sigma^T \omega(t) \times x),
\]

according to equation (3.1), given by \( \tilde{\omega}(t) = -\Sigma^T \omega(t) = (-\alpha(t) v(t), \omega_3(t))^T \). This non-uniqueness is also observed in cryo-EM, where the reconstruction of the rotations is only possible up to a global orthogonal transformation [9].

In general, one may, of course, also have other solutions; in the extreme case of a spherically symmetric attenuation coefficient, for example, the attenuation projection data contains no information on the orientation of the object and every rotation \( R \) will fulfill equation (3.9). However, for a sufficiently asymmetric object, we can expect a unique solution \( \tilde{\nu} \) up to sign so that \( \Psi_E \) is constant, which is also indicated by our numerical tests, see section 4.

### 3.2.2. Reconstruction of the cylindrical radius of the angular velocity

So far we have explained how to recover the cylindrical parameters \( t \mapsto v(t) \) (or equivalently \( t \mapsto \varphi(t) \)) and \( t \mapsto \omega_3(t) \). For the full reconstruction of the motion, we will still need to estimate the cylindrical radius \( t \mapsto \alpha(t) \), which can be done by using propositions 3.8 below. Note that we go directly to the third derivative of equation (3.9), since the second derivative provides no new information.

**Proposition 3.8.** Let \( u \in C_c^\infty(\mathbb{R}^3; \mathbb{R}) \), \( \tilde{\mathcal{J}} \) be the reduced attenuation mapping of a rigid motion of \( u \) in Fourier space. Let further \( R \in C^4(\mathbb{R}; SO(3)) \), \( t \in \mathbb{R} \) and \( \omega \in C^3(\mathbb{R}; \mathbb{R}^3) \) be the angular velocity corresponding to \( R \) and let \( \sigma(t) = \varphi'(t) \). Then, for all \( t \in \mathbb{R} \) such that \( \alpha(t) \neq 0 \) and \( \sigma(t) \neq -\omega_3(t) \), we have

\[
A_0(t, \mu) + A_0(t, \mu) \alpha^2(t) + A_1(t, \mu) \frac{\alpha'(t)}{\alpha(t)} = 0 \quad \text{for all } \mu \in \mathbb{R},
\]

\[ (3.14) \]
where
\[
A_0(t, \mu) = \frac{1}{2} \mu(\omega_3 + \sigma) \left[ \mu^2 \omega_3 (\omega_3 - \sigma) D^1[\tilde{J}] (t, \mu \nu) \right] [v^+, v^+] \\
+ 2 \mu D^2[\tilde{J}] (t, \mu \nu) [v^+, \omega_3 \sigma v - \omega_3' v^+] + 2D^3[\tilde{J}] (t, \mu \nu) [\omega_3^2 v^+ + \omega_3' v] \\
- \mu (3 \omega_3 - \sigma) \partial_3 D^1[\tilde{J}] (t, \mu \nu) [v^+, v^+] + 2 \partial_3 D^1[\tilde{J}] (t, \mu \nu) [v^+],
\]
\[
A_{02}(t, \mu) = \frac{1}{2} \mu(\omega_3 + \sigma) D^1[\tilde{J}] (t, \mu \nu) [v^+],
\]
\[
A_1(t, \mu) = \frac{1}{2} (\omega_3 + \sigma) \left[ \mu \omega_2 D^2[\tilde{J}] (t, \mu \nu) [v^+, v^+] - \omega_3 D^1[\tilde{J}] (t, \mu \nu) [v] - \partial_3 D^1[\tilde{J}] (t, \mu \nu) [v^+] \right].
\]

Here, we omitted for the sake of readability the arguments of the functions \( v, v^+, \sigma, \) and \( \omega_3 \) and their derivatives: they are always to be evaluated at the time \( t \).

**Proof.** The proof is provided in appendix.

It is possible to recover for every time \( t \in \mathbb{R} \) the parameter \( \alpha(t) \) using propositions 3.8. Since equation (3.14) holds for every \( \mu \in \mathbb{R} \), we can consider it as an overdetermined linear system for \( \omega^2(t) \) and \( \frac{\varphi'(t)}{\omega_3(t)} \), see section 4.

**Example.** Apparently, the coefficients \( A_0, A_{02}, \) and \( A_1 \) in propositions 3.8 vanish if \( \omega_3 = -\sigma \), in which case equation (3.14) becomes trivial and thus a reconstruction of \( \alpha \) from this equation impossible.

We want to study this in the simple case of an angular velocity \( \omega \) for which the cylindrical coordinate \( \alpha \) and the third component \( \omega_3 \) are constant functions. Then, the critical condition \( \sigma(t) = \varphi'(t) = -\omega_3 t \) occurs everywhere if we choose \( \varphi(t) = -\omega_3 t \).

In this case, the rotation \( R \) can be explicitly calculated by solving equation (3.1) and we find with the normalization \( R(0) = 1 \) that
\[
R(t) = \begin{pmatrix}
\cos(\omega_3 t) & -\sin(\omega_3 t) & 0 \\
-\sin(\omega_3 t) \cos(\alpha t) & -\cos(\omega_3 t) \cos(\alpha t) & \sin(\alpha t)
\end{pmatrix}.
\]

We therefore have
\[
P \left( e_3 \times (R(t)^T R(t)) e_3 \right) = \sin(\alpha(t - s)) \begin{pmatrix}
\cos(\omega_3 s) \\
-\sin(\omega_3 s)
\end{pmatrix},
\]
which means that the common line property, equation (3.9), upon which the whole reconstruction is based, reduces to
\[
\tilde{J} \left( s, \frac{\lambda}{t - s} \sin(\alpha(t - s)) \begin{pmatrix}
\cos(\omega_3 s) \\
-\sin(\omega_3 s)
\end{pmatrix} \right) = \tilde{J} \left( t, \frac{\lambda}{s - t} \sin(\alpha(t - s)) \begin{pmatrix}
-\cos(\omega_3 t) \\
\sin(\omega_3 t)
\end{pmatrix} \right)
\]
for all \( \lambda \in \mathbb{R} \). Since the term \( \sin(\alpha(t - s)) \) can be absorbed into the variable \( \lambda \), this equation contains no information about \( \alpha \). Therefore, it is not possible to recover this special motion completely from the common line property alone.
4. Numerical simulation

Finally, we want to give a proof of principle that the results in section 3 allow us under some conditions to reconstruct the motion correctly. To this end, we choose an attenuation coefficient without obvious symmetries to prevent multiple solutions of equation (3.11) for the components $v$ and $\omega_3$ of the angular velocity $\omega = (\alpha v_1, \alpha v_2, \omega_3)^T$. We thus consider for the points $P_1 = (1, 1/2, -1)^T$, $P_2 = (-1/2, 1, 1)^T$, $P_3 = (0, -1, 1/2)^T$ and the diagonal matrix $D = \text{diag}(\sqrt{2}, 1, 1)$ as an example the attenuation coefficient

$$u(x) = \prod_{i=1}^{3} |x - P_i|^2 e^{-\frac{1}{4} |Dx|^2}$$

(4.1)

and define the motion $(T, R)$ by the arbitrary choice

$$T(t) = \begin{pmatrix} \cos(6t) \cos(12t) \\ \cos(6t) \sin(12t) \\ \sin(t) \end{pmatrix} \quad \text{and} \quad R(t) = R_3(\psi_3(t))R_2(\psi_2(t))R_3(\psi_3(t)).$$

(4.2)

where we pick for the rotation angles the functions

$$\psi_1(t) = t^2, \quad \psi_2(t) = t + 3t^3, \quad \text{and} \quad \psi_3(t) = \sin(3t).$$

(4.3)

Here, we denoted by $R_2(\phi)$ and $R_3(\phi)$ the rotations by the angle $\phi \in \mathbb{R}$ around the standard basis vectors $e_2$ and $e_3$, respectively:
\[
R_2(\phi) = \begin{pmatrix}
\cos(\phi) & 0 & \sin(\phi) \\
0 & 1 & 0 \\
-\sin(\phi) & 0 & \cos(\phi)
\end{pmatrix}
\quad \text{and} \quad
R_3(\phi) = \begin{pmatrix}
\cos(\phi) & -\sin(\phi) & 0 \\
\sin(\phi) & \cos(\phi) & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

Since \( R_2(\phi)^T R_2(\phi)x = e_2 \times x \) and \( R_3(\phi)^T R_3(\phi)x = e_3 \times x \) for all \( x \in \mathbb{R}^3 \), we can write the derivative of the rotation \( R \) in the form
\[
R(t)^T R'(t)x = [\psi_3'(t)R_3(\psi_3(t)) + \psi_2'(t)R_2(\psi_2(t))]T R_2(\psi_2(t)) R_3(\psi_3(t))
+ \psi_1'(t) [R_1(\psi_1(t)) R_2(\psi_2(t)) R_3(\psi_3(t))]x
= \psi_3'(t)e_3 \times x + \psi_2'(t)(R_3(\psi_3(t))T e_2) \times x + \psi_1'(t)(R_3(\psi_3(t))^T R_2(\psi_2(t))^T e_3) \times x
\]
for every \( x \in \mathbb{R}^3 \). This gives us according to equation (3.1) the angular velocity
\[
\omega(t) = \begin{pmatrix}
\psi_2'(t) \sin(\psi_3(t)) - \psi_1'(t) \sin(\psi_2(t)) \cos(\psi_3(t)) \\
\psi_2'(t) \cos(\psi_3(t)) + \psi_1'(t) \sin(\psi_2(t)) \sin(\psi_3(t)) \\
\psi_1'(t) + \psi_2'(t) \cos(\psi_2(t))
\end{pmatrix},
\]
which we can write in cylindrical coordinates to obtain the parameters \( \alpha, \varphi, \) and \( \omega_3 \) as defined in equation (3.2); see figure 7 for a plot of the graphs of these functions.

We discretize the transformed function \( x \mapsto u(C_3 + R(t)(x - C_3 + T(t))) \), where \( C_3 \) denotes the center of \( u \), by evaluating it at the points
\[
(j_1, j_2, j_3) \delta_3, \quad \text{where} \quad (j_1, j_2, j_3) \in \{-512, \ldots, 511\}^3 \times \{-256, \ldots, 255\} \quad \text{and} \quad \delta_3 = 0.05,
\]
and calculate the attenuation mapping \( J[T, R] \) of \( u \) by projecting along the third direction at every time step
\[
\ell \delta_3, \quad \text{where} \quad \ell \in \{0, \ldots, 9999\} \quad \text{and} \quad \delta_0 = 0.0001.
\]

Some attenuation projection images are depicted in figure 4.

Following the reconstruction formulas presented in propositions 3.3, 3.7, and 3.8, we will attempt to reconstruct the first two components \( T_1 \) and \( T_2 \) of the translation \( T \) and the rotation \( R \), parametrized by the functions \( \alpha, \varphi, \) and \( \omega_3 \), from the data \( J[T, R] \). In view of the instability of this reconstruction algorithm due to the differentiations of the reduced attenuation mapping \( \tilde{J} \) and of the recovered versions of the functions \( \varphi \) and \( \omega_3 \) with respect to time, see propositions 3.7 and 3.8, we consider here as a basic test of the practicability of the reconstruction only the ideal case of data without noise and with a sufficiently fine discretization so that we avoid the problems caused by numerical differentiation. Under more realistic settings, appropriate regularization methods will have to be used to mitigate this ill-posedness, but this is out of the scope of this article.

For the reconstruction of \( T \) and \( \omega \) (from which we can then recover the rotation \( R \) by solving the differential equation in equation (3.1) up to the arbitrary initial rotation \( R(0) \)), we proceed for each time step \( t = \ell \delta_3 \) as follows.

(i) We determine the center \( C_2(t) \) of the attenuation mapping \( J[T, R](t, \cdot) \) and read off the first two components of the displacement \( T(t) - T(0) \) via equation (3.6), see figure 5. Moreover, we calculate from this the reduced attenuation mapping \( \tilde{J} \) via equation (3.7).
(ii) If we had already found the correct value \( \varphi(t) \) for the cylindrical angle of \( \omega(t) \), then we could interpret equation (3.11) as a linear regression problem to determine the third component \( \omega_3(t) \) of \( \omega(t) \):
\[ g(t, \mu, \phi(t)) \omega_3(t) = h(t, \mu, \phi(t)) \]

with

\[ g(t, \mu, \phi) = D^1[\tilde{\mathcal{J}}(t, \mu \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}) \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \end{pmatrix}] \quad \text{and} \quad h(t, \mu, \phi) = \partial_\mu \tilde{\mathcal{J}}(t, \mu \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}). \]

We therefore define \( \tilde{\omega}_3(t, \phi) \) for each \( \phi \in [0, \pi) \) as the minimum point of the least squares functional

\[ W_3 : \omega_3 \mapsto \sum_{j=-J_0}^{J_0} |g(t, j\delta_x, \phi)w_3 - h(t, j\delta_x, \phi)|^2 \tag{4.5} \]

with the (for noiseless data arbitrary) cutoff \( J_0 = 50 \), which is given explicitly by

\[ \tilde{\omega}_3(t, \phi) = \frac{\sum_{j=-J_0}^{J_0} g(t, j\delta_x, \phi)h(t, j\delta_x, \phi)}{\sum_{j=-J_0}^{J_0} g^2(t, j\delta_x, \phi)}; \]

so that we have

\[ \omega_3(t) = \tilde{\omega}_3(t, \phi(t)). \tag{4.6} \]

The values of \( \tilde{\mathcal{J}} \) and its derivatives, which appear in this formula, can be nicely interpolated (with the help of the Whittaker–Shannon interpolation formula, for example) as they are Fourier transforms of very rapidly decreasing functions.

(iii) To recover the cylindrical angle \( \phi(t) \), we remark that the minimum of the functional \( W_3 \) from equation (4.5), which is for every choice of \( \phi \) attained at the point \( \tilde{\omega}_3(t, \phi) \), should theoretically become zero at the correct value \( \phi(t) \). We therefore reconstruct \( \phi(t) \) as the minimum point of the functional

\[ \Phi : [0, \pi) \to [0, \infty), \phi \mapsto W_3(\tilde{\omega}_3(t, \phi)) = \sum_{j=-J_0}^{J_0} |g(t, j\delta_x, \phi)\tilde{\omega}_3(t, \phi) - h(t, j\delta_x, \phi)|^2. \]

Since \( \Phi \) is in general non-convex, but only a one-dimensional function, we choose to simply minimize it on the sufficiently fine grid \( \{m\delta_\phi \mid m \in \{0, \ldots, 2047\} \} \), \( \delta_\phi = \frac{\pi}{2048} \), by evaluating it at all points (and by additionally refining the grid in the vicinity of the
minimum point to improve the accuracy), see figure 6 for a sketch of the graph of $\Phi_t$ at the time step $50\delta_t$.

From the so reconstructed value of $\phi(t)$, we then calculate the value of $\omega_3(t)$ via equation (4.6).

(iv) To obtain $\alpha$, we consider equation (3.14) as an overdetermined linear system (one equation for each value $\mu \in \{j\delta_x \mid j \in \{-J_0, \ldots, J_0\}\}$ for $\alpha^2(t)$ and $\frac{\alpha'(t)}{\alpha(t)}$, where the coefficients $A_{0b}$, $A_{02}$, and $A_1$ can be explicitly calculated with the values of $\phi$ and $\omega_3$ obtained so far.

This means, we reconstruct the vector $(\alpha^2(t), \frac{\alpha'(t)}{\alpha(t)})$, and thus $\alpha(t) > 0$, as the minimum point of the functional

$\mathbb{R}^2 \to \mathbb{R}, \ V \mapsto |A(t)V - b(t)|^2$

with the matrix $A(t) \in C^{(2J_0+1)\times 2}$ and the vector $b(t) \in C^{2J_0+1}$ given by

$A_{j1}(t) = A_{02}(t, j\delta_x)$,
$A_{j2}(t) = j\delta_x A_1(t, j\delta_x)$, and
$b_j(t) = -A_0(t, j\delta_x)$

for $j \in \{-J_0, \ldots, J_0\}$.Explicitly, this minimum point is calculated as a solution of the normal equation

$A(t)^TA(t)V = A(t)^Tb(t)$.

The results of this reconstruction algorithm (applied to the motion specified in equations (4.2) and (4.3)) for the parameters $\alpha$, $\phi$, and $\omega_3$ of the angular velocity $\omega$ are shown in figures 7 and 8. They agree up to the expected error caused by the discretization well with the exact values calculated from equation (4.4).
4.1. Comparison to ASPIRE

There exists a large number of software tools or software applications that are used in the field of single-particle cryo-EM, which also take on the task of orientation estimation. One of them is the ASPIRE package [12], which provides common-line based algorithms for orientation assignment. Here, we want to draw a comparison between (the MATLAB® version of) ASPIRE and our algorithm using the simulated data defined via equations (4.1), (4.2) and (4.3). Since we are not familiar with all the details of ASPIRE, some of the obtained results might still be improvable.

The orientation estimation done by ASPIRE, which is based on [15], takes as an input a stack of projection images and gives back the corresponding rotation matrices. We have calculated the translations with our algorithm and used the reduced attenuation projection images independent of the translations as an input for ASPIRE, so that only the rotations are left to be determined. As it is designed to solve problems from cryo-EM, the orientation estimation algorithm of ASPIRE expects projection images from different viewing angles (class averages) that are not very similar to each other. As the temporal step size in our generated data is very small, we only used every 105th image for the reconstruction. This seemed to yield the best results.
In figure 9 the reconstructed rotation matrices are depicted via their Euler angle representations. Finally, in figure 7 the corresponding angular velocity parameters are shown next to the exact as well as our reconstructed values. We used a Gaussian filter for computing the derivatives of the rotation matrices from ASPIRE, which explains the deviations of the reconstructed values close to the boundaries of the time interval. The fact that our algorithm takes, in contrast to ASPIRE, the temporal coherence of the data into account is clearly an advantage.

Figure 8. Graphs of the absolute errors between the exact and the reconstructed values of the parameters $\alpha$, $\varphi$, and $\omega_3$. (Errors less than $10^{-6}$ are not shown.)

Figure 9. Graphs of the Euler angles $\psi_1$, $\psi_2$ and $\psi_3$ together with the values reconstructed with ASPIRE. The errors of the reconstructed values of $\psi_1$ and $\psi_3$ appear due to the non-uniqueness in the calculation of the Euler angles for small $\psi_2$ (gimbal lock).

In figure 9 the reconstructed rotation matrices are depicted via their Euler angle representations. Finally, in figure 7 the corresponding angular velocity parameters are shown next to the exact as well as our reconstructed values. We used a Gaussian filter for computing the derivatives of the rotation matrices from ASPIRE, which explains the deviations of the reconstructed values close to the boundaries of the time interval. The fact that our algorithm takes, in contrast to ASPIRE, the temporal coherence of the data into account is clearly an advantage.
However, being built for problems from cryo-EM, the ASPIRE package is designed to deal with a lot of noise, while our algorithm is based on numerical differentiation and is therefore very sensitive to faulty data. An algorithm that combines the best of both worlds would be ideal for data from optical microscopy of trapped specimens. We plan to introduce appropriate regularization methods to our algorithm, which should improve performance with noisy data while still utilizing the time component.

5. Conclusions and outlook

Tomographic reconstruction of a trapped object has the intrinsic problem that the object’s rotation is typically not as smooth and well-defined (and therefore known) as applied during tomographic data acquisition in other settings. Instead, the object undergoes a more irregular motion described as time-dependent translations and rotations.

In the present work we deliver a first step into the direction of tomographic reconstruction of optically or acoustically trapped particles. We assume imaging at low numerical aperture of absorptive samples with amplitude contrast, which means that the attenuation projection describes the imaging process reasonably well.

For this case, we demonstrate—by explicit reconstruction—how the motional parameters can be inferred, wherever permitted uniquely. However, not all parameters are uniquely retrievable, for example, the component of the translation in the direction the projection images are taken cannot be seen, or, in case of symmetries of the sample, we encounter fake solutions of our equations for the angular velocity (in the extreme case of a spherically symmetric object, no information on the motion is available). More uniqueness studies are on the way.

The mathematical problem formulation reveals similarities to single-particle cryo-EM. However, in our trapping application, we have the additional information that there is a continuous relation between the time and the orientation of the object. In the future, the proposed motion estimation will be tested on video data acquired from biological samples held in optical or acoustic traps, see figure 10. Moreover, it will be necessary to study corrections or alternative approaches required when going from attenuation projection images to optical images. Image formation of amplitude- or phase-samples in a microscope may significantly differ from attenuation projections, as explained in [1, 2].
Acknowledgments

The authors are supported by the Austrian Science Fund (FWF) with SFB F68 (‘Tomography across the Scales’), project F6804-N36 (‘Quantitative Coupled Physics Imaging’), project F6806-N36 (‘Inverse Problems in Imaging of Trapped Particles’), and project F6807-N36 (‘Tomography with Uncertainties’). The authors thank Mia Kvåle Løvmo and Benedikt Thalhammer for providing data concerning the noise levels in the optical trapping experiments.

Appendix

The main ingredient of our results for motion estimation in section 3 are based on calculating higher order derivatives of composed functions. For this we will use a tensor notation.

A.1. Derivatives in tensor notation

Let

\[ f : \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{C}, \]

\[ (t, k) \rightarrow f(t, k) \]

and

\[ g : \mathbb{R} \rightarrow \mathbb{R}^2, \]

\[ t \mapsto \begin{pmatrix} g_1(t) \\ g_2(t) \end{pmatrix}. \]

In this paper we use derivatives of up to order 3 of the composed function

\[ h : \mathbb{R} \rightarrow \mathbb{C}, \]

\[ t \mapsto h(t) := f(t, g(t)). \]

These derivatives will be expressed via a tensor notation:

- We denote by \( D^i[f](t, \kappa) : \mathbb{R}^2 \times \mathbb{R}^2 \times \cdots \times \mathbb{R}^2 \rightarrow \mathbb{C} \) the derivative of order \( i = 0, 1, 2, \ldots \) of the function \( k \mapsto f(t, k) \) with respect to the variable \( k \) for fixed \( t \in \mathbb{R} \) at a point \( \kappa \in \mathbb{R}^2 \).

We write the evaluation of the tensor \( D^i[f](t, \kappa) \) in the form

\[ D^i[f](t, \kappa)[v_1, v_2, \ldots, v_i]. \]

- The time derivatives of the composed function \( t \mapsto h(t) \) up to order three can then be expressed as follows:

\[ h'(t) = D^1[f](t, g(t))[g'(t)] + \partial f(t, g(t)), \]

\[ h''(t) = D^2[f](t, g(t))[g'(t), g''(t)] + 2 \partial f(t, g(t)) [g''(t)] + \partial^2 f(t, g(t)), \]

\[ h'''(t) = D^3[f](t, g(t))[g'(t), g''(t), g'''(t)] + 3 \partial f(t, g(t)) [g''(t), g'''(t)] + \partial^3 f(t, g(t)). \]

(A.1)
Note that for $i = 1, 2$ the tensor notation simplifies to the usual matrix vector multiplications:
\[
\mathbf{D}^1[f](t, g(t))[g'(t)] = (\nabla_k[f](t, g(t)), g'(t)) ,
\]
\[
\mathbf{D}^2[f](t, g(t))[g'(t), g'(t)] = g'(t)^T \nabla_k^2[f](t, g(t)) g'(t).
\]

### A.2. Higher order expansion of angular velocities

As we would like to expand the common line property in lemma 3.6 in a Taylor polynomial, we first calculate the relevant Taylor coefficients of the arguments of the function $J$ therein.

**Lemma A.1.** Let $R \in C^1(\mathbb{R}; SO(3))$, $t \in \mathbb{R}$ and $\omega \in C^3(\mathbb{R}; \mathbb{R}^3)$ be the angular velocity corresponding to $R$. As in equation (3.2), we write $\omega = (\alpha \omega, \omega \omega)^T$ with $\alpha \in C^3(\mathbb{R}; \mathbb{R}^+)$, $\nu \in C^3(\mathbb{R}; \mathbb{S}^1)$, and $\omega_3 \in C^3(\mathbb{R}; \mathbb{R})$. Then, we have

\[
\frac{1}{t-\delta} P(e_3 \times R(s)^T R(t)e_3) = \sum_{j=0}^{3} a_j(s)(t-s)^j + \bar{a}(s,t),
\]
\[
\frac{1}{s-t} P(e_3 \times R(t)^T R(s)e_3) = \sum_{j=0}^{3} b_j(s)(t-s)^j + \bar{b}(s,t)
\]

with remainder terms $a, b$ fulfilling $\lim_{t\rightarrow s} \frac{a_j(s)}{(s-t)^j} = \lim_{t\rightarrow s} \frac{b_j(s)}{(s-t)^j} = 0$ and with the coefficients

\[
a_0 = \alpha \nu,
\]
\[
a_2 = \omega \nu (\alpha + (\alpha \nu)^\prime),
\]
\[
a_6 = -\omega (\alpha + (\alpha \nu)^\prime) \nu - 2 \omega \nu (\alpha \nu)^\prime + (\alpha \nu)^\prime \nu + (\alpha \nu)^\prime \nu,
\]
\[
a_{24} = -\omega (\alpha + (\alpha \nu)^\prime) \nu - 2 \omega \nu (\alpha \nu)^\prime + (\alpha \nu)^\prime \nu + (\alpha \nu)^\prime \nu,
\]
\[
b_0 = \alpha \nu,
\]
\[
b_2 = -\omega \nu (\alpha + (\alpha \nu)^\prime),
\]
\[
b_6 = -\omega (\alpha + (\alpha \nu)^\prime) \nu - 2 \omega \nu (\alpha \nu)^\prime + (\alpha \nu)^\prime \nu + (\alpha \nu)^\prime \nu,
\]
\[
b_{24} = (\alpha + (\alpha \nu)^\prime) \nu - 2 \omega \nu (\alpha \nu)^\prime + (\alpha \nu)^\prime \nu + (\alpha \nu)^\prime \nu.
\]

**Proof.** According to Taylor’s theorem, the coefficients $a_j$ and $b_j$, $j = 0, 1, 2, 3$, are given by

\[
a_j = \frac{1}{(j+1)!} P(e_3 \times R^{(j+1)} e_3)
\]
\[
b_j = \frac{1}{(j+1)!} P(e_3 \times (R^{(j+1)})^T Re_3).
\]

To calculate the derivatives of $R$, we abbreviate the matrix-valued function $R^TR'$ by $B$ and calculate from the identity $R' = RB$ the higher derivatives

\[
R^TR'' = B^2 + B',
\]
\[
R^TR''' = B^3 + 2BB' + B'B + B'',
\]
\[
R^TR'''' = B^4 + 3B^2 B' + 2BB'B + 3BB'' + B'B^2 + 3B'' + 3(B')^2 + B''B + B'''.
\]
Plugging in equation (3.1) to write $Bx = \omega \times x$ and applying the identity $f \times (g \times h) = (f, h) g - (f, g) h$ for $f, g, h \in \mathbb{R}^3$, we find

$$a_0 = P(\omega),$$

$$2a_1 = P(\omega_3 e_3 \times \omega + \omega'),$$

$$6a_2 = P\left( -|\omega|^2 \omega + 2\omega_3 e_3 \times \omega' + \omega_3 e_3 \times \omega + \omega'' \right),$$

$$24a_3 = P\left( -|\omega|^2 \omega_3 e_3 \times \omega + 3(\omega_3 \omega_3' \omega - \omega_3^2 \omega' - (\omega_3, \omega') \omega - 2(\omega, \omega') \omega + 3\omega_3 e_3 \times \omega''
+ (\omega_3^2 \omega' - \omega_3 \omega'' \omega - |\omega|^2 \omega') + 3\omega_3 e_3 \times \omega' + \omega_3' e_3 \times \omega + \omega'' \right)
= P\left( -|\omega|^2 \omega_3 e_3 \times \omega + 2\omega_3 \omega_3' \omega - 2\omega_3^2 \omega' - 5(\omega, \omega') \omega + 3\omega_3 e_3 \times \omega''
- |\omega|^2 \omega' + 3\omega_3 e_3 \times \omega' + \omega_3' e_3 \times \omega + \omega'' \right),$$

$$b_0 = P(\omega),$$

$$2b_1 = P\left( -\omega_3 e_3 \times \omega + \omega' \right),$$

$$6b_2 = P\left( -|\omega|^2 \omega - \omega_3 e_3 \times \omega' - 2\omega_3' e_3 \times \omega + \omega'' \right),$$

$$24b_3 = P\left( |\omega|^2 \omega_3 e_3 \times \omega + (\omega_3 \omega_3' \omega - \omega_3^2 \omega' - (\omega, \omega') \omega - 2(\omega, \omega') \omega - \omega_3 e_3 \times \omega''
+ 3(\omega_3^2 \omega' - \omega_3 \omega'' \omega - |\omega|^2 \omega') + 3\omega_3 e_3 \times \omega' + 3\omega_3' e_3 \times \omega + \omega'' \right)
= P\left( |\omega|^2 \omega_3 e_3 \times \omega - 2\omega_3 \omega_3' \omega - 2\omega_3^2 \omega' - 3(\omega, \omega') \omega - \omega_3 e_3 \times \omega''
- 3|\omega|^2 \omega' - 3\omega_3' e_3 \times \omega' - 3\omega_3' e_3 \times \omega + \omega'' \right),$$

where we can use the antisymmetry $B = -B^T$ for the calculation of the coefficients $b_i$.

Inserting $P(\omega) = \alpha \nu$, we end up with equation (A.3).

\[\square\]

### A.3. Derivation of the reconstruction for the cylindrical radius of the angular velocity

Finally, we will use the expansion in lemma A.1 to get the equation in third order of the common line property, lemma 3.6, from which we want to reconstruct the cylindrical radius $\alpha$ of the angular velocity $\omega$ as defined in equation (3.2).

#### Proof of proposition 3.8

By taking three derivatives of equation (3.9) with respect to $t$ and evaluating at $t = s$, using the Taylor polynomials in equation (A.2) of the arguments of $\tilde{J}$, we find with equation (A.1) (omitting the argument $s$ in the coefficients $a$ and $b$)

$$D^3[\tilde{J}](s, \lambda a_0)[\lambda a_1, \lambda a_2] + 6D^2[\tilde{J}](s, \lambda a_0)[\lambda a_1, \lambda a_2] + 6D[\tilde{J}](s, \lambda a_0)[\lambda a_1]$$

$$= D^3[\tilde{J}](s, \lambda b_0)[\lambda b_1, \lambda b_2] + 6D^2[\tilde{J}](s, \lambda b_0)[\lambda b_1, \lambda b_2] + 6D[\tilde{J}](s, \lambda b_0)[\lambda b_1]$$

$$+ 3\partial_1 D^3[\tilde{J}](s, \lambda b_0)[\lambda b_1, \lambda b_2] + 6\partial_1 D^2[\tilde{J}](s, \lambda b_0)[\lambda b_2]$$

$$+ 3\partial_2 D^3[\tilde{J}](s, \lambda b_0)[\lambda b_1] + \partial_{\lambda b_0} \tilde{J}(s, \lambda b_0).$$

(4.4)

We calculate some of the terms separately by using our expressions for the coefficients $a$ and $b$ in equation (A.3) (also omitting the argument $s$ in the function $\alpha$, $\nu$, $\omega_3$ and their derivatives):

- The third derivatives of $\tilde{J}$ yield

$$D^3[\tilde{J}](s, \lambda a_0)[\lambda a_1, \lambda a_2, \lambda a_3] - D^3[\tilde{J}](s, \lambda b_0)[\lambda b_1, \lambda b_1, \lambda b_1]$$

$$= \frac{1}{4} \lambda^3 \left( \omega_3^3 D^3[\tilde{J}](s, \lambda \alpha)[\omega^4, \omega^4, \omega^4] + 3\omega_3 D^3[\tilde{J}](s, \lambda \alpha)[\omega^4, (\omega v)'] \right).$$

(A.5)
• For the second derivatives of \( \tilde{J} \), we find
\[
6D^2[\tilde{J}](\lambda, \lambda_0)[\lambda a_0, \lambda_2] - 6D^2[\tilde{J}](\lambda, \lambda_0)[\lambda b_1, \lambda b_2]
\]
\[
= \lambda^2 \omega_3 \left( D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, (\alpha \alpha)''] - (\alpha^2 + \omega_3^2) D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, \alpha \alpha] \right)
\]
\[
+ \frac{1}{2} \lambda^2 \omega_3 \left( \omega D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, (\alpha \alpha)''] - \omega D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, \alpha \alpha] \right)
\]
\[
+ \frac{3}{2} \lambda^2 \left( \omega D^2[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha')'] + \omega D^2[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha')', (\alpha \alpha)'''] \right).
\]
(A.6)

• For the first derivatives of \( \tilde{J} \), we get with
\[
\nu := -(\alpha^2 + \omega_3^2) \omega_3 \alpha \alpha^+ - 2 \omega_3 \alpha \alpha' - 2 \omega_3 (\alpha \alpha)''
\]
\[
- (\alpha \alpha', (\alpha \alpha')' + \omega_3 \alpha \alpha + 2 \omega_3 (\alpha \alpha^+))''
\]
\[
+ (\alpha^2 + \omega_3^2) (\alpha \alpha')' + 3 \omega_3 (\alpha \alpha^+) + 2 \omega_3 (\alpha \alpha^+),
\]

that
\[
6D^1[\tilde{J}](\lambda, \lambda_0)[\lambda a_0] - 6D^1[\tilde{J}](\lambda, \lambda_0)[\lambda b_2] = \frac{1}{2} \lambda D^1[\tilde{J}](\lambda, \lambda \alpha)[\nu].
\]
(A.7)

• And for the mixed derivatives of \( \tilde{J} \), we have
\[
3 \partial_3 D^2[\tilde{J}](\lambda, \lambda_0)[\lambda b_1, \lambda b_2] + 6 \partial_3 D^1[\tilde{J}](\lambda, \lambda_0)[\lambda b_2]
\]
\[
+ 3 \partial_3 D^1[\tilde{J}](\lambda, \lambda_0)[\lambda b_1] + \partial_m \tilde{J}(\lambda, \lambda_0)
\]
\[
= \frac{3}{4} \lambda \left( \omega \partial_3 D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, \alpha \alpha] \right.
\]
\[
- 2 \omega_3 \partial_3 D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, (\alpha \alpha)' + \partial_3 D^2[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha)', (\alpha \alpha)'' \right]
\]
\[
+ \lambda \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha)'' - 2 \omega_3 (\alpha \alpha^+) - \omega_3 (\alpha \alpha)'' - (\alpha^2 + \omega_3^2) \alpha \alpha]
\]
\[
+ \frac{3}{2} \lambda \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha')' - \omega_3 \alpha \alpha^+] + \partial_m \tilde{J}(\lambda, \lambda \alpha).
\]
(A.8)

Looking only at the terms including a second derivative of \( \alpha \), we see that they cancel out:
\[
\lambda^2 \omega_3 \alpha \alpha'' D^2[\tilde{J}](\lambda, \lambda \alpha)[\nu, \nu] + \lambda \omega \alpha \alpha'' D^1[\tilde{J}](\lambda, \lambda \alpha)[\nu] - \lambda \omega D^1[\tilde{J}](\lambda, \lambda \alpha)[\nu] = 0,
\]
(A.9)

since we get from taking the derivative of equation (3.12) with respect to \( \lambda \) that
\[
\alpha \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[\nu] = \omega_3 \alpha \alpha D^2[\tilde{J}](\lambda, \lambda \alpha)[\nu, \nu] + \omega D^1[\tilde{J}](\lambda, \lambda \alpha)[\nu].
\]
(A.10)

Furthermore, taking the time derivative of equation (3.12), we arrive at
\[
\partial_3 \tilde{J}(\lambda, \lambda \alpha) + \lambda \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha)']
\]
\[
= \lambda \omega_3 D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+] + \lambda \omega_3 \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+]
\]
\[
+ \omega_3 \partial_3 D^1[\tilde{J}](\lambda, \lambda \alpha)[(\alpha \alpha')'] + \lambda^2 \omega_3 D^2[\tilde{J}](\lambda, \lambda \alpha)[\alpha \alpha^+, \alpha \alpha^+].
\]

Then, a second derivative with respect to time gives us the identity
\[
\partial_m \tilde{J}(s, \lambda \alpha) + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda^2 \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + 3 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad = \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + 2 \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda^2 \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + (\alpha')', (\alpha')'.
\] (A.11)

Subtracting now the identity equation \(A.11\) from equation \(A.4\), we find, by plugging in the expressions in equations \((A.5), (A.6), (A.7)\) and \((A.8)\) for the derivatives and using that the terms involving \(\alpha''\) vanish according to equation \(A.9\), the relation

\[
\frac{1}{4} \lambda^3 \left( \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \right) \\
\quad - \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \frac{1}{2} \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \frac{1}{2} \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \frac{1}{2} \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] \\
\quad + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - (\alpha'' + \omega_3 \alpha''(\alpha')' - (\alpha', (\alpha')') + (\alpha')') \\
\quad + \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] - (\alpha'' + \omega_3 \alpha''(\alpha')' - (\alpha', (\alpha')') + (\alpha')') \\
\quad + \frac{1}{2} \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + (\alpha'' + \omega_3 \alpha''(\alpha')' - (\alpha', (\alpha')') + (\alpha')') \\
\quad + \lambda \omega_3 D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + (\alpha'' + \omega_3 \alpha''(\alpha')' - (\alpha', (\alpha')') + (\alpha')').
\]

We sort the terms with respect to the term \(\alpha'\) and write the equation in the form

\[
A_0(s, \lambda \alpha) + \alpha^2 A_{02}(s, \lambda \alpha) + A_1(s, \lambda \alpha) \lambda \alpha' + A_{12}(s, \lambda \alpha)(\lambda \alpha')^2 = 0. \tag{A.12}
\]

- The coefficient \(A_{12}\) of \((\alpha')^2\) is then given by

\[
A_{12}(s, \mu) = -\frac{1}{4} \mu \omega_3 D^1[\tilde{J}](s, \mu \alpha)[(\alpha')'] \left[ v^+, v, v \right] - \frac{1}{4} \omega_3 D^1[\tilde{J}](s, \mu \alpha)[(\alpha')'] \left[ v, v^+, v \right] + \frac{1}{4} \partial_\alpha D^1[\tilde{J}](s, \mu \alpha)[(\alpha', v)'] \left[ v, v, v \right]. \tag{A.13}
\]

By taking another derivative with respect to \(\lambda\) of equation \(A.10\), we find the identity

\[
\alpha^2 \partial_\alpha D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] = 2 \omega_3 \alpha \partial_\alpha D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + \lambda \omega_3 \alpha D^1[\tilde{J}](s, \lambda \alpha)[(\alpha')'] + \lambda \omega_3 \alpha (\alpha')' + \lambda \omega_3 \alpha (\alpha')'.
\]

Comparing this with equation \(A.13\), we see that

\[
A_{12} = 0.
\]

- The coefficient \(A_1\) of \(\alpha'\) is seen to be
\[
A_1(s, \mu) = -\frac{1}{2} \mu^2 \omega_3 D^3 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v, v'] \} + \frac{1}{2} \mu \omega_3^2 D^3 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \}
- \frac{1}{2} \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v, v'] \} - \frac{1}{2} \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \}
- \frac{1}{2} \mu \omega_3 \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v, v^+] \} + \frac{1}{2} \mu \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v, v'] \}
- \omega_3 \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \} + \frac{1}{2} \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v] \}.
\]

If we compare this with the time derivative of the derivative of equation (3.11) with respect to \( \mu \), that is, the time derivative of

\[
\partial_\nu D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v] \} = \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v] \} + \omega_3 D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}.
\]

which is

\[
\partial_\nu D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v] \} + \mu \partial_ \nu D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v'] \}
= \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v] \} + \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v'] \} + \omega_3 D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}
+ \omega_3 \partial_\nu D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}.
\]

then we see that some of the coefficients cancel each other and we are left with

\[
A_1(s, \mu) = \frac{1}{2} \mu \omega_3 D^3 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \} + \frac{1}{2} \mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \} - \frac{1}{2} \omega_3^2 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}
- \frac{1}{2} \omega_3 \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \} + \frac{1}{2} \partial_\nu D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}.
\]

We rewrite this in the form

\[
A_1(s, \mu) = \frac{1}{2} \left[ \mu \omega_3 D^3 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \} + \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \} \right]
+ \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \} \left( v^+ \right) - \omega_3 \partial_\nu \left[ \tilde{\mathcal{J}}(s, \mu \nu) \right] \left[ \omega_3 v^+ + v' \right].
\]

Remarking that \(|v|^2 = 1\) and thus \((v, v') = 0\), we have a scalar function \( \sigma \) such that

\[
v' = \sigma v^+.
\]

With this, \( A_1 \) becomes (using \((v^+) = (v') = -\sigma v))

\[
A_1(s, \mu) = \frac{1}{2} \left[ \omega_3 + \sigma \right] \left[ \mu \omega_3 D^3 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v^+] \} - \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \} - \partial_\nu \left[ \tilde{\mathcal{J}}(s, \mu \nu) \right] \left[ \omega_3 v^+ + v' \right].
\]

- The coefficient \( A_{02} \) of \( \alpha^2 \) consists of the terms

\[
A_{02}(s, \mu) = -\mu \omega_3 D^2 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+, v] \} - \frac{1}{2} \mu \omega_3 D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v^+] \}
+ \frac{1}{2} \mu \partial_\nu D^1 \{ \tilde{\mathcal{J}}(s, \mu \nu) [v] \}.
\]
With equation (A.10) this becomes
\[ A_{02}(s, \mu) = \frac{1}{2} \mu \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) \omega_3 \sigma^+ + \sigma^+ v' = \frac{1}{2} \mu (\omega_3 + \sigma) \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) v^+ \].

Finally, the coefficient \(A_0\) is given by
\[
A_0(s, \mu) = \frac{1}{4} \mu^3 \omega_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+, v^+] - \frac{1}{4} \mu^3 \omega_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v', v']
- \frac{1}{2} \mu^2 \omega_3 \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [\omega_3 v^+ - v', (v^+)^']
- \frac{1}{2} \mu^2 \omega_3 \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [\omega_3 v^+ + v', v^+]
+ \frac{1}{2} \mu \omega_3 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [\omega_3 v - (v^+)^']
- \frac{3}{4} \mu^2 \omega_3 \partial_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+]
+ \frac{1}{4} \mu^2 \partial_3 \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [v^+ - 2 \omega_3 v^+, v^+]
- \mu \omega_3 \partial_3 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [(v^+)^' - \omega_3 v]
+ \frac{1}{2} \mu \partial_3 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [v' + \omega_3 v^+].
\]

Writing again \(v' = \sigma v^+\), this becomes
\[
A_0(s, \mu) = \frac{1}{4} \mu^3 \omega_3 (\omega_3^2 - \sigma^2) \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+, v^+]
- \frac{1}{2} \mu^2 \omega_3 (2 \omega_3^2 + \omega_3^2 + \omega_3^2 - \sigma^2) \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [v^+, v]
- \frac{1}{2} \mu^2 (\omega_3 + \sigma) \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+]
+ \frac{1}{4} \mu (\omega_3 + \sigma) \partial_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+]
+ \mu \omega_3 (\omega_3^2 - \sigma) \partial_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+] + \frac{1}{2} \mu (\omega_3 + \sigma) \partial_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+].
\]

Subtracting equation (A.14), we can get rid of one term:
\[
A_0(s, \mu) = \frac{1}{4} \mu (\omega_3 + \sigma) \left[ \mu^2 \omega_3 (\omega_3^2 - \sigma) \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+, v^+]\right]
+ 2 \mu \partial_3 \mathbf{D}^2 [\tilde{\mathcal{J}}](s, \mu) [v^+, \omega_3 \sigma v - \omega_3 v^+]
+ 2 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [\omega_3^2 v^+ + \omega_3^2 v^+]
+ \mu (\omega_3 + \sigma) \partial_3 \mathbf{D}^3 [\tilde{\mathcal{J}}](s, \mu) [v^+, v^+]
+ 2 \partial_3 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [v^+] + 2 \partial_3 \mathbf{D}^1 [\tilde{\mathcal{J}}](s, \mu) [v^+].
\]

\[\square\]

**ORCID iDs**

Peter Elbau @ https://orcid.org/0000-0001-5894-5793
Monika Ritsch-Marte @ https://orcid.org/0000-0002-5945-546X
Otmar Scherzer @ https://orcid.org/0000-0001-9378-7452
Denise Schmutz @ https://orcid.org/0000-0002-3773-1750
References

[1] Arridge S R 1999 Optical tomography in medical imaging Inverse Problems 15 R41–93
[2] Arridge S R and Schotland J 2009 Optical tomography: forward and inverse problems Inverse Problems 25 123010
[3] Bracewell R 1956 Strip integration in radio astronomy Aust. J. Phys. 9 198
[4] Coifman R R, Shkolnisky Y, Sigworth F J and Singer A 2010 Reference free structure determination through eigenvectors of center of mass operators Appl. Comput. Harmon. Anal. 28 296–312
[5] Goncharov A B 1988 Methods of integral geometry and recovering a function with compact support from its projections in unknown directions Acta Appl. Math. 11 213–22
[6] van Heel M 1987 Angular reconstitution: a posteriori assignment of projection directions for 3d reconstruction Ultramicroscopy 21 111–23
[7] Helgason S 1999 The Radon Transform (Progress in Mathematics vol 5) 2 edn (Boston, MA: Birkhäuser) p xiv + 188
[8] Jones P H, Maragò O M and Volpe G 2015 Optical Tweezers. Principles and Applications (Cambridge: Cambridge University Press)
[9] Lamberg L 2008 Unique recovery of unknown projection orientations in three-dimensional tomography Inverse Problems Imaging 2 547–75
[10] Müller P, Schürmann M and Guck J 2015 The theory of diffraction tomography (arXiv:1507.00466)
[11] Natterer F 2001 The Mathematics of Computerized Tomography (Classics in Applied Mathematics vol 32) (Philadelphia, PA: SIAM) p xviii + 222
[12] Singer A and Shkolnisky Y 2019 Aspire algorithms for single particle reconstruction http://spr.math.princeton.edu/
[13] Singer A and Shkolnisky Y 2019 Viewing direction estimation in cryo-EM using synchronization SIAM J. Imaging Sci. 6 1088–110
[14] Thalhammer G, Steiger R, Meinschad M, Hill M, Bernet S and Ritsch-Marte M 2011 Combined acoustic and optical trapping Biomed. Opt. Express 2 2859–70
[15] Wang L, Singer A and Wen Z 2013 Orientation determination of cryo-EM images using least unsquared deviations SIAM J. Imaging Sci. 6 2450–83