ICES REPORT 12-26

July 2012

Non-uniform Fourier Transforms for Multi-dimensional Rotational Correlations

by

C. Bajaj, B. Bauer, R. Bettadapura, A. Vollrath

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: C. Bajaj, B. Bauer, R. Bettadapura, A. Vollrath, Non-uniform Fourier Transforms for Multi-dimensional Rotational Correlations, ICES REPORT 12-26, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, July 2012.

NON-UNIFORM FOURIER TRANSFORMS FOR MULTI-DIMENSIONAL ROTATIONAL CORRELATIONS

CHANDRAJIT BAJAJ[1], BENEDIKT BAUER[2], RADHAKRISHNA BETTADAPURA[3], AND ANTJE VOLLRATH[4]

Abstract. The rigid-body correlation problem seeks the the rigid-body transformation $(\mathbf{R}, \mathbf{t})$, $\mathbf{R} \in SO(3)$, $\mathbf{t} \in \mathbb{R}^3$ that maximizes the correlation between a pair of input scalar-valued functions $A : \mathbb{R}^3 \mapsto \mathbb{C}$ and $B : \mathbb{R}^3 \mapsto \mathbb{C}$. Exhaustive solutions to the rigid-body correlation problem take advantage of the equispaced fast Fourier transform to achieve either translational speedups (over $\mathbb{R}^3$) or rotational speedups (over $SO(3)$). We present PF*corr*, a new exhaustive solution, based on the non-equispaced $SO(3)$ Fourier transform, to the rigid-body correlation problem; unlike previous solutions, ours achieves a combination of translational and rotational speedups while encouraging, but not requiring, the presence of non-equispaced grids. PF*corr* can be straightforwardly applied to a variety of problems in protein structure prediction and refinement that involve correlations under rigid-body motions of the protein. Additionally, we show how it applies, along with an appropriate flexibility model, to analogs of the above problems in which the flexibility of the protein is relevant.

AMS Subject Classification:

Key words: Fast Fourier methods, Motion Groups, Match and Fit, Exhaustive search, Fast correlations

1. Introduction. The task of evaluating correlations is central to computational structural biology. The correlation problem—finding the best relative orientation between a pair of "entities"—manifests itself concretely in various stages of biological structure elucidation, from molecular replacement [33], where the entities in question are interatomic vector maps known as Patterson maps, to protein-protein docking [2], where the entities are a pair of crystal or NMR structures.

The problem of rigid-body correlation is one of optimization: given a pair of real- or complex scalar-valued functions A and B defined on a grid , find the rigid-body transformation of B that *optimizes* its correlation with A. The primary contribution of this work is PF*corr*, a Fourier-based solution to the rigid-body correlation problem that addresses two major deficiencies in existing Fourier-based approaches; we discuss these deficiencies in the following section, which also provides a brief overview of present work in this area.

Our secondary contribution has to do with *flexible* correlations, a problem complementary to rigid-body correlations. Biomolecules undergo flexible deformations in solvent, and thus most problem domains that begin by assuming that the protein is rigid go on to consider its flexibility. The flexible correlation problem can be solved with a suitable parametrization of the space of flexible motions of the protein, after which each element of that space is just a rigid entity, conducive to rigid-body correlations. We present for the first time an exhaustive algorithm that, with a suitable biomolecular flexibility model, can also perform flexible correlations.

[1]Computational Visualization Center, Department of Computer Sciences and The Institute of Computational Engineering and Sciences, The University of Texas at Austin, 1 University Station C0200, Austin, Texas 78712, USA. Email: bajaj@cs.utexas.edu

[2]Institute of Computational Mathematics, University of Luebeck, Luebeck, Germany. Email: bauer@cls.uni-luebeck.de

[3]Computational Visualization Center, Department of Mechanical Engineering, The University of Texas at Austin, 1 University Station C0200, Austin, Texas 78712, USA. Email: brrk@utexas.edu

[4]Institute of Computational Mathematics, TU Braunschweig, Pockelsstr 14, 38106 Braunschweig, Germany. Email: a.vollrath@tu-bs.de

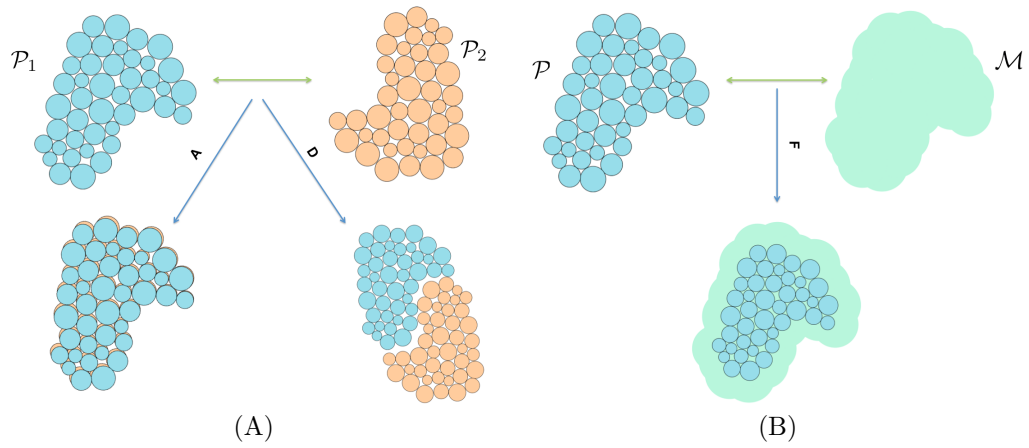


FIG. 1.1. *Sample applications of the correlation algorithm we develop in this work. (A) Protein-protein docking and alignment. (B) Protein-density map fitting.*

2. Related and prior work. The vast number of extant solutions to the rigid-body correlation problem can be distinguished by a few basic approaches. Iterative approaches vary in sophistication, ranging from a simple version of steepest ascent [24] to more powerful techniques such as Powell optimization [44]. Most such approaches result in locally optimal solutions that, depending on the initial guess, may or may not be close to the globally optimal correlation. They are thus usually used along with an exhaustive approach that provides the requisite initial guess.

Feature-based approaches compute and correlate reduced representations of A and B. An early example of a feature-based approach is the method of vector quantization [46], in which sets of vectors are used to represent A and B. A similar approach is geometric hashing [18], whereby critical features on both of A and B are hashed into a table of values, and a score—related to the correlation score—measures the match between A and B for a particular relative orientation. Feature-based approaches, used in docking [34] and fitting [43], result in improved performance due to the reduced search space, at the possible expense of poor resolution scaling.

Exhaustive approaches attempt to compute the global maximum of the correlation between A and B. Early exhaustive methods to solve the rigid-body correlation problem relied on a simple and profound insight: by the Fourier cross-correlation theorem, computing a discretized, uniformly spaced version of the rigid-body correlation is equivalent to computing $\mathcal{O}(1)$ forward and inverse FFTs [14], turning an effectively impractical $\mathcal{O}(N^6 N_{rot})$ algorithm into a feasible one that scales as $\mathcal{O}(N^3 \log N N_{rot})$, where N is the maximum grid-size and N_{rot} the number of rotations. Combined with a fast, multi-platform implementation of the FFT [10, 37], this technique, which results in speedups over the translational search space $\mathbb{R}^3$, has found widespread use in rigid-body docking and fitting routines [2, 21, 25, 45, 47]. For completeness, we mention the $SE(3) = \mathbb{R}^3 \times SO(3)$ Fourier transform introduced and developed by Chirikjian et. al [6], which, while applying to topics ranging from the workspace density of robotic manipulators to the conformational statistics of macromolecules [7], has so far not been applied to the focus of this work, i.e., correlations over the space of rigid-body or flexible motions of a protein.

FFT-based techniques are very efficient at surveying the space of translations;

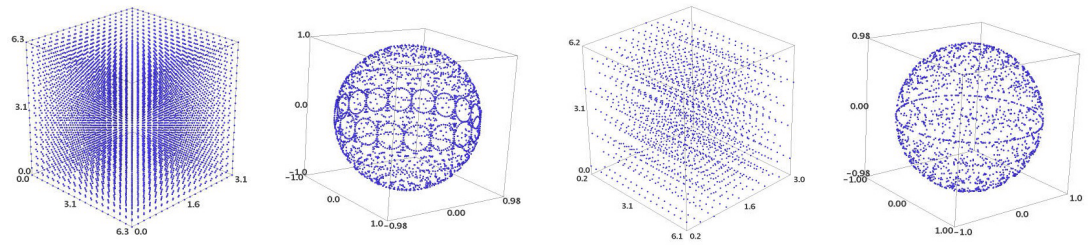


FIG. 2.1. *The uniformity conundrum. (A) A uniform z-y-z Euler Angular grid with angular resolution* $= 20°$ *leads to (B) a non-uniform sampling of* $SO(3)$*, with very high angular resolution in certain regions and "holes" in certain others. By contrast, (C) a highly non-uniform Euler angular grid leads to (D) a more uniform sampling of* $SO(3)$*. (C) and (D) were obtained by the techniques in [22]; they contain fewer samples, exhibit a separation very close to the required angular resolution of* $20°$*, and are highly uniform with respect to certain metrics(Section 7). Can a rotationally efficient rigid-body correlation search algorithm aspire to use the samples in (D)?*

unfortunately, though, this space consists for the most part of obviously poor correlations. In most problem domains, the range of plausible solutions exists in a highly localized band of translations. On the other hand, the band of plausible relative rotations spans the range 0 to 2π for each translation. It is thus more important to be able to efficiently sample rotational space. Rotational speedups depend on representing A and B in a basis more amenable to rotational sampling. In Kovacs and Wriggers [17], Kovacs et. al [16], and Garçon et. al [11], that basis is the basis of functions on the unit sphere $\mathbb{S}^2$, i.e., the family of spherical harmonic functions $Y_\ell^m(\theta, \phi)$, whereas in the work of Ritchie [30, 32], a radial basis function $R_k^\ell(r)$ related to the Gaussian accompanies the spherical basis. Like their translational counterparts, rotational speedups compute a multiple exponential sum, or an FFT; unlike translational speedups, the FFT is computed on a uniformly spaced grid of z-y-z Euler angles.

Current exhaustive Fourier-based techniques suffer from two drawbacks.

Drawback 1. The first drawback relates to local refinement: depending as they do on the equispaced FFT, exhaustive techniques cannot be gracefully used to refine existing solutions. Say we wish to improve a docking pose, obtained using a translational FFT speedup with a grid size of $x\mathring{A}$. If we redo the experiment with a grid size of $\frac{x}{2}$, the (3D) FFT becomes eight times as expensive, but more importantly, it spends much of its time at points on the new grid already excluded by the initial experiment. A similar argument applies to rotational speedups; in both these approaches, the concept of a local refinement is largely absent.

Drawback 2. The second, related, drawback, relates to the question of uniform sampling in rotational space. While sampling in translational space is straightforward, involving Cartesian grids with uniform, possibly differing grid-sizes in each independent direction, the notions of "uniformity" and "direction" do not translate easily to the rotational space $SO(3)$. In particular, equispaced Euler angular grids do not result in equispaced $SO(3)$ samples (Figure 2.1). Due to this, rotational FFT-based techniques, despite their *raison d'être*, are destined to oversample certain regions of $SO(3)$ while leaving others wholly unexamined[1].

[1] Of course, translational FFT-based techniques can afford to uniformly, albeit inefficiently, sample $SO(3)$.

2.1. Proteins and flexibility. Due to the vastness of the space of flexible motions, protein flexibility can be practically dealt with by (A) conducting all-atomistic local searches, as in the case of molecular dynamical algorithms [12, 13, 19, 36, 38], (B) Building a coarse-grained representation of the protein, also known as a domain decomposition [1, 9, 26, 35], or (C) A combination of the strategies in (A) and (B) [41, 42, 48].

Domain-based approaches have so far lacked a search scheme that takes advantage of the translational or rotational speedups that FFT-based approaches can afford. This has to do with the issue of focusing: in uniform FFT-based techniques, there is no way to restrict the search space to a small area of interest that can be occupied by a single domain rather than the entire protein. By contrast, searching over the entire space for each domain is both time-consuming and results in spurious and geometrically implausible false positives, and sifting through these grows rapidly inefficient as the number of domains increases. This is also why domain-based flexibility algorithms such as those in Topf et. al [40, 41] and Trabuco et. al [42] prefer Monte-Carlo-based or steepest-ascent-based search schemes.

2.2. Primary contributions.

1. **Rigid-body correlations.** We address the drawbacks in Section 2 with a pair of rotationally exhaustive, non-equispaced techniques to compute rigid-body correlations. Using the concept of Wigner-d-Chebyshev and Chebyshev-exponential transforms first articulated by one of the authors in Potts et.al [27], we convert Equation 3.1 into a multiple exponential sum, which we then compute using a combination of non equispaced SO(3) transforms [27] and non equispaced FFTs [15]. The resulting family of techniques, which we call PF*corr*, has the following properties:
   - **Sampling robust.** The technique is capable of efficiently computing correlations over arbitrary samples in $\mathbb{R}^3 \times SO(3)$.
   - **Compatible.** It can be used along with existing equispaced FFT-based techniques.
   - **General.** It unifies the rotationally-exhaustive paradigms in Kovacs and Wriggers, Ritchie, and Ritchie et. al [11, 17, 30, 32].

   PF*corr* thus provides an alternative to existing rigid-body correlation techniques.
2. **Flexible correlations.** The second half of this work presents an algorithm that uses PF*corr* to explore correlations in multi-domain search spaces. The non-uniformity inherent to PF*corr* implies that these correlations can be focused in a specific subset of $\mathbb{R}^3 \times SO(3)$, while its exhaustive nature guarantees that it is not sensitive to local optima.

We believe that the above properties, along with its speed, make PF*corr* a realistic and in many ways preferable alternative to existing correlation search schemes.

2.3. Secondary contributions. One of the two halves of PF*corr* depends on a mixed radial/spherical basis (introduced in the following section), which in turn depends on looking up translation matrix (T-Matrix) entries for $SO(3)$. The high complexity of computing the T-matrix entries means that they often have to be precomputed and stored. We outline an efficient algorithm, based on polynomial update rules, to compute T-Matrix entries that, while not obviating the need for precomputation and storage, has nevertheless a lower complexity than existing algorithms.

Finally, this work also aims to be a self-contained overview of correlation techniques that depend on expressing the input scalar valued functions in terms of ro-

tationally invariant bases. In particular, we prove all relevant properties inherent to our mathematical framework, even ones that may be seen as elementary and/or well-known.

Note: We defer most multi-line proofs to the Appendix.

3. Background. Let $A, B : \mathbb{R}^3 \mapsto \mathbb{C}$ be a pair of scalar-valued functions. We define the rigid-body correlation problem as follows.

DEFINITION 3.1. ***Rigid-body correlation problem.*** *Let $A : \mathbb{R}^3 \mapsto \mathbb{C}$ and $B : \mathbb{R}^3 \mapsto \mathbb{C}$ be a pair of scalar-valued functions. Define*

$$C(\mathbf{R}_i, \mathbf{t}_j) = \int_{\mathbb{R}^3} A(\mathbf{x}) B(\mathbf{R}_i \mathbf{x} + \mathbf{t}_j) d\mathbf{x},$$

$$i \in \{1 \dots N_{rot}\}, j \in \{1 \dots N_{trans}\} \tag{3.1}$$

as the rigid-body correlation between A and B for a given set $S = \{(\mathbf{R}_i, \mathbf{t}_j)\}, R_i \in SO(3), \mathbf{t}_j \in \mathbb{R}^3$ of rigid-body transformations. The rigid-body correlation problem is to maximize $C(\mathbf{R}_i, \mathbf{t}_j)$ over the set S.

The rigid-body correlation problem is a non-convex geometric optimization problem, and the several problem domains in computational biology to which it applies can be distinguished by their choice of A and B. In protein-protein docking, for instance, A and B are affinity functions that represent a relevant property, such as shape or electrostatics, of the underlying protein; in protein-density map fitting, A is a blurred representation of the atoms of the protein, while B is (usually) the density map itself.

All approaches to the rigid-body correlation problem begin by representing A and B in terms of appropriate orthogonal basis functions. For instance, FFT-based techniques that produce speedups over translations $\mathbf{t}_j$ of B are obtained by expanding A and B in Fourier bases. Similarly, speedups over the space of rotations $\mathbf{R}_i$ of B are effected by first expanding A and B in *spherical* Fourier bases. We adopt an instance of the latter approach in this work, and as a preliminary, introduce the basis functions in question in the following discussion.

3.1. Orthogonal radial and spherical basis functions. Let the spherical coordinates of $\mathbf{x} \in \mathbb{R}^3$ be $(r, \mathbf{u})$, where $r \in \mathbb{R}^+$ is the radial coordinate and $\mathbf{u} = (\theta, \phi) \in [0, \pi) \times [0, 2\pi]$ the polar and azimuthal components of $\mathbb{S}^2$ respectively. The weighted Laguerre polynomials are radial bases for complex scalar-valued functions on $\mathbb{S}^2$.

DEFINITION 3.2. ***Weighted Laguerre polynomials.*** *For $r \in \mathbb{R}_0^+$, $\ell, k \in \mathbb{N}_0$, $k > \ell$, the weighted Laguerre polynomials $R_k^\ell : \mathbb{R}^+ \to \mathbb{R}$ are given by*

$$R_k^\ell(r) = \beta_{k\ell} \mathrm{e}^{-\frac{r^2}{2\lambda}} \frac{r^\ell}{\sqrt{\lambda}} L_{k-\ell-1}^{\ell+\frac{1}{2}} \left(\frac{r^2}{\lambda} \right)$$

with

$$\beta_{k\ell} = \sqrt{\frac{2(k-\ell-1)!}{\lambda^{\frac{3}{2}} \Gamma(k+\frac{1}{2})}}$$

where L_k^ℓ are the Laguerre polynomials [39], and $\lambda \in \mathbb{R}^+$ dictates the rate of decay of the basis function.

LEMMA 3.3. *For $r \in \mathbb{R}_0^+,\ \ell, k \in \mathbb{N}_0,\ k > \ell$, the functions $R_k^\ell(r)$ satisfy*

$$\int_0^\infty R_k^\ell(r) R_n^\ell(r) r^2 \mathrm{d}r = \delta_{k,n}.$$

The weighted Laguerre polynomials have been described in detail in the quantum mechanics literature, where they arise naturally as radial bases to solutions of certain harmonic oscillators [5]. They have also recently been used in Ritchie [31] in the context of 6 dimensional rigid-body docking.

The family of spherical harmonic functions, indexed by degree ℓ and order m, form a spherical basis for complex scalar-valued functions in $\mathbb{S}^2$.

DEFINITION 3.4. ***Spherical harmonics.*** *For any $\ell \in \mathbb{N}_0$ and $m = -\ell, \ldots \ell$ the spherical harmonics of degree ℓ are defined as*

$$Y_\ell^m(\xi) = \sqrt{\frac{2\ell+1}{4\pi}} P_\ell^{|m|}(\cos\theta) \mathrm{e}^{\mathrm{i}m\phi}$$

where $P_\ell^m : [-1,1] \to \mathbb{R}$ are associated Legendre polynomials*, cf. [39], that arise as the derivatives of ordinary Legendre polynomials $P_\ell(x)$.*
The spherical harmonics satisfy the orthogonality relation

$$\int_{\mathbb{S}^2} Y_\ell^m(\xi) \overline{Y_{\ell'}^{m'}(\xi)} \, \mathrm{d}\xi = \delta_{\ell\ell'}\delta_{mm'}. \tag{3.2}$$

Combining each of the orthogonality relations, we see that the functions $R_k^\ell(r) Y_\ell^m(\mathbf{u})$ for $k, \ell \in \mathbb{N},\ k > l \geq |m|$ are orthonormal with respect to the inner product:

$$\begin{aligned}\langle R_k^\ell(r) Y_\ell^m(\mathbf{u}), R_{k'}^{\ell'}(r) Y_{\ell'}^{m'}(\mathbf{u}) \rangle &= \int_0^\infty R_k^\ell(r) R_{k'}^{\ell'}(r) r^2 \mathrm{d}r \int_{\mathbb{S}^2} Y_\ell^m(\mathbf{u}) \overline{Y_{\ell'}^{m'}(\mathbf{u})} \mathrm{d}\mathbf{u} \\ &= \delta_{k,k'}\delta_{\ell,\ell'}\delta_{m,m'}. \end{aligned} \tag{3.3}$$

3.2. Multi-basis framework. As a first step in solving the rigid-body correlation problem in Equation 3.1, PF*corr* represents A and B in terms of orthogonal basis functions. PF*corr* offers two distinct choices of basis functions:

1. **Mixed radial/spherical bases.** Following Ritchie [32], it can use Laguerre polynomials R_k^ℓ for the radial basis and the spherical harmonic functions Y_ℓ^m for the spherical basis, or,
2. **Pure spherical basis.** Following Kovacs [17] and Garçon [11], it can use Y_ℓ^m for the spherical basis on each radial slice r.

The orthogonality of the radial and spherical bases functions results in the following two expansions.

Mixed radial/spherical basis expansion. The mixed radial/spherical representation of a scalar valued function $A : \mathbb{R}^3 \mapsto \mathbb{C}$ is given by

$$A(\boldsymbol{x}) = A(r, \mathbf{u}) = \lim_{L\to\infty} \sum_{k=1}^{L} \sum_{\ell=0}^{k-1} \sum_{m=-\ell}^{l} \hat{a}_{k\ell m} R_k^\ell(r) Y_\ell^m(\mathbf{u}) \tag{3.4}$$

with coefficients

$$\hat{a}_{k\ell m} = \int_{\mathbb{R}^+} \int_{\mathbb{S}^2} A(r, \mathbf{u}) \overline{R_k^\ell(r) Y_\ell^m(\mathbf{u})} r^2 \mathrm{d}\mathbf{u}\mathrm{d}r. \tag{3.5}$$

Pure spherical basis expansion. The pure spherical representation of a scalar valued function $A : \mathbb{R}^3 \mapsto \mathbb{C}$ for a given radial coordinate r is given by

$$A_r(\mathbf{u}) = \lim_{L\to\infty} \sum_{l=0}^{L} \sum_{m=-l}^{l} \hat{a}_{\ell m}(r) Y_\ell^m(\mathbf{u}) \tag{3.6}$$

with coefficients

$$\hat{a}_{\ell m}(r) = \int_{\mathbb{S}^2} A_r(\mathbf{u}) \overline{Y_\ell^m(\mathbf{u})} \mathrm{d}\mathbf{u}, \tag{3.7}$$

where $A_r(\mathbf{u}) = A(r, \mathbf{u})$.

For computational purposes, the term L in Equations 3.4 and 3.7 is set to values between 20 and 30, depending on the application.

3.3. Rotations in $\mathbb{R}^3$. An orthogonal 3×3 matrix with unit determinant represents a rotation in $\mathbb{R}^3$. The *special orthogonal group* SO(3) is the collection of these matrices

$$\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I},\ |\mathbf{R}| = 1\},$$

equipped with the usual group action, and neutral and inverse elements. The Z-Y-Z *Euler angle decomposition* of a rotation $\mathbf{R} \in \mathrm{SO}(3)$ is the representation

$$\mathbf{R} = \mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_Z(\alpha)\, \mathbf{R}_Y(\beta) \mathbf{R}_Z(\gamma),$$

with angles $\alpha, \gamma \in [0, 2\pi)$ and $\beta \in [0, \pi]$, and the Y-axis and Z-axis rotation matrices

$$\mathbf{R}_Y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}, \qquad \mathbf{R}_Z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The space of Euler Angles (α, β, γ) parametrizes the space $SO(3)$ of rotations.

The space of square integrable functions in $SO(3)$ is denoted $\mathrm{L}^2(\mathrm{SO}(3))$ and defined via the standard inner product

$$\langle f, g\rangle = \int_0^{2\pi} \int_0^{\pi} \int_0^{2\pi} f(\alpha, \beta, \gamma) \overline{g(\alpha, \beta, \gamma)} \sin\beta \, \mathrm{d}\gamma \, \mathrm{d}\beta \, \mathrm{d}\alpha.$$

A convenient orthogonal basis for $\mathrm{L}^2(\mathrm{SO}(3))$ are the Wigner-D functions $D_\ell^{m,n}$ with degree ℓ and orders m, n with $\max\{|m|, |n|\} \le \ell$ are given by the explicit expression

$$D_\ell^{m,n}(\alpha, \beta, \gamma) = \mathrm{e}^{-\mathrm{i}m\alpha}\, d_\ell^{m,n}(\cos\beta)\, \mathrm{e}^{-\mathrm{i}n\gamma}$$

where $d_\ell^{m,n}$ are the Wigner-d functions

$$d_\ell^{m,n}(x) = \varepsilon \left(\frac{s!(s+\mu+\nu)!}{(s+\mu)!(s+\nu)!} \right)^{1/2} 2^{-\frac{\mu+\nu}{2}} (1-x)^{\frac{\mu}{2}} (1+x)^{\frac{\nu}{2}} P_{\ell-L_*}^{(\mu,\nu)}(x), \tag{3.8}$$

$P_{\ell-L_*}^{(\mu,\nu)}(x)$ are the Jacobi polynomials and

$$\mu = |n-m|, \qquad \nu = |n+m|, \qquad \varepsilon = \begin{cases} 1, & \text{if } m > n, \\ (-1)^{n-m}, & \text{if } m \le n. \end{cases}$$
$$L_* = \max\{|m|, |n|\}, \quad s = \ell - L_*,$$

Note that $d_\ell^{m,n}$ is a polynomial of degree ℓ if $m+n$ is even. Otherwise, it is a polynomial of degree $\ell-1$ times a factor of $(1-x^2)^{1/2}$.

The Wigner-D functions satisfy the orthogonality condition

$$\langle D_\ell^{m,n}, D_{\ell'}^{m',n'} \rangle = \frac{8\pi^2}{2\ell+1}\delta_{\ell,\ell'}\delta_{m,m'}\delta_{n,n'}.$$

3.4. Rotating basis expansions of scalar-valued functions. The expansion $\hat{a}_{klm}$ or $\hat{a}_{lm}$ of a scalar valued function can be rotated by rotating the basis functions used to generate it. In particular, since the spherical harmonic functions Y_ℓ^m are rotationally invariant, i.e.

$$Y_\ell^n(\boldsymbol{Ru}) = \sum_{m=-\ell}^{\ell} Y_\ell^m(\boldsymbol{u}) D_\ell^{nm}(\boldsymbol{R}), \qquad \text{for } |m| \le \ell, \boldsymbol{u} \in \mathbb{S}^2, \boldsymbol{R} \in \mathrm{SO}(3), \tag{3.9}$$

we have, for $\mathbf{R} \in SO(3)$,

$$A(\boldsymbol{Rx}) = A(r, \boldsymbol{Ru}) = \sum_{k=1}^{\infty}\sum_{\ell=0}^{k-1}\sum_{m,n=-\ell}^{\ell} \hat{a}_{k\ell m} D_\ell^{m,n}(\mathbf{R}) R_k^\ell(r) Y_\ell^m(\mathbf{u}).$$

A similar result holds for the radial-basis independent coefficients $\hat{a}_{lm}$.

3.5. Translating basis expansions of scalar-valued functions. Let $\mathbf{t}_z = z\mathbf{e}_z$ be a translation along the z-axis. Then, following Ritchie [30] and Danos and Maximon [8]

$$A(\mathbf{x}+\mathbf{t}_z) = A_{\mathbf{t}_z}(r,\mathbf{u}) = \sum_{k=1}^{L}\sum_{l=0}^{k-1}\sum_{m=-l}^{l}\sum_{n=-l}^{l}\sum_{j=1}^{\infty}\sum_{h=0}^{j-1} \hat{a}_{klm} T_{jh,kl}^{|n|}(\Delta z) R_j^h(r) Y_h^n(\mathbf{u}) \tag{3.10}$$

where $T_{jh,kl}^{|n|}$ are the SO(3) translation matrix entries for the translation Δz. Note that the translation matrices apply only to mixed radial-spherical basis expansions $\hat{a}_{klm}$; for pure spherical basis expansions, the coefficients $\hat{a}_{lm}$ for each radial slice r have to be recomputed after each translation $\mathbf{t} \in \mathbb{R}^3$.

Ritchie derives an analytic expression for the translation coefficients $T_{kl,k'l'}^{|m|}$ in the case of Laguerre basis functions [30].

DEFINITION 3.5. ***Translation matrix entries for*** $SO(3)$***.*** *The translation coefficients are expressed as [30]*

$$T_{k'l',kl}^{|m|}(t) = \tag{3.11}$$

$$\exp(-z^2/4\lambda) \sum_{n=|l-l'|}^{l+l'} A_n^{ll'|m|} \sum_{j=0}^{k-l+k'-l'-2} C_j^{kl,k'l'} M! (z^2/4\lambda)^{n/2} L_M^{(n+1/2)}(z^2/4\lambda), \tag{3.12}$$

where

$$M = j + (l + l' - n)/2,$$

$$C_j^{k'l',kl} = \sum_{j=0}^{k-l-1} \sum_{j'=0}^{k'-l'-1} \delta_{n,j+j'} X_{klj} X_{k'l'j'},$$

$$X_{klj} = \left[\frac{(k-l-1)!(1/2)_k}{2}\right]^{1/2} \frac{(-1)^{k-l-j-1}}{j!(k-l-j-1)!(1/2)_{l+j+1}},$$

$(1/2)_n = 1/2 \cdot (1/2+1) \cdot (1/2+2) \ldots (1/2+n-1)$ *is the Pochhammer symbol,*

$$L_k^\alpha(x) = \sum_{j=0}^{k} \binom{k+\alpha}{k-j} \frac{(-x)^j}{j!} \text{ are generalized Laguerre polynomials,}$$

$$A_n^{l'l|m|} = (-1)^{n+l'-l)/2+m}(2n+1)\left[(2l'+1)(2l+1)\right]^{1/2} \begin{pmatrix} l' & l & n \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l' & l & n \\ m & -m & 0 \end{pmatrix},$$

and $\begin{pmatrix} a & b & c \\ \alpha & \beta & \gamma \end{pmatrix}$ *denotes the Wigner 3-j symbol given by*

$$\begin{pmatrix} a & b & c \\ \alpha & \beta & \gamma \end{pmatrix} =$$

$$(-1)^{a-b-\gamma}\sqrt{\Delta(abc)}\sqrt{(a+\alpha)!(a-\alpha)!(b+\beta)!(b-\beta)!(c+\gamma)!(c-\gamma)!} \sum_{t=t_{\min}}^{t_{\max}} \frac{(-1)^t}{x},$$

$$\Delta(abc) = \frac{(a+b-c)!(a-b+c)!(-a+b+c)!}{(a+b+c+1)!},$$

$$x = t!(c-b+t+\alpha)!(c-a+t-\beta)!(a+b-c-t)!(a-t-\alpha)!(b-t+\beta)!,$$

where $t_{\min} = \max(0, b-c-\alpha, a+\beta-c)$ *and* $t_{\max} = \min(a+b-c, a-\alpha, b+\beta)$.

Naively computing T-Matrix entries for fixed k, l, k', l', m takes $\mathcal{O}(L^3 N_t)$ steps, where N_t is the number of translations. The overall complexity is thus $L^5 \cdot \mathcal{O}(L^3 N_t) = \mathcal{O}(L^8 N_t)$.

4. Rigid-body correlations. Radial/spherical harmonic expansions can be used to compute rigid-body correlations. Let A and B be scalar-valued functions, and let B undergo rotations $\mathbf{R}$ relative to A. We are interested in the pure rotational correlation $C(\mathbf{R}) = \int_{\mathbb{R}^3} A(\mathbf{x})\overline{(B(\mathbf{R}\mathbf{x}))}d\mathbf{x}$, where the overbar represents complex conjugation[2]. The following two lemmas can be established, respectively, for mixed-basis coefficients $\hat{a}_{klm}, \hat{b}_{klm}$ and pure spherical basis coefficients $\hat{a}_{lm}, \hat{b}_{lm}$:

LEMMA 4.1.

$$C(\mathbf{R}) = \sum_{k=1}^{L} \sum_{\ell=0}^{k-1} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell}^{\ell} (-1)^m \hat{a}_{k\ell-m}(-1)^{m'} \overline{\hat{b}_{k\ell-m'}} D_\ell^{m,m'}(\mathbf{R}). \tag{4.1}$$

[2]The conjugation is used to simplify algebraic manipulations, and is otherwise redundant.

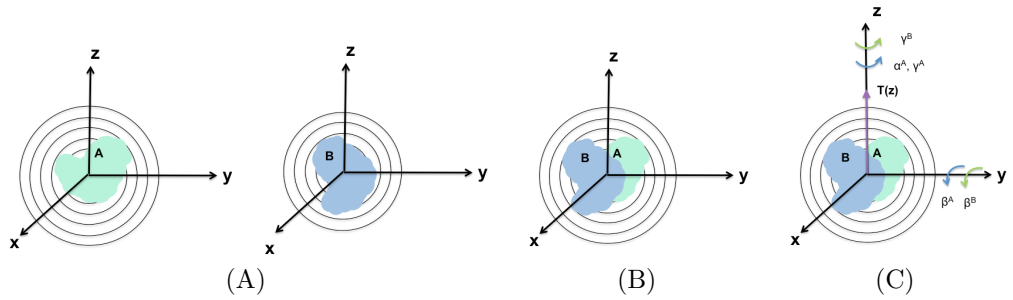


FIG. 4.1. *Schematic of the rigid-body correlation search scheme introduced in this work. Here A, B are two complex or real scalar-valued functions. (A) Initial positions of A and B in different coordinate frames. (B) A and B are translated to share the same origin. (C) A rigid-body motion comprising a single translation z along the z-axis and five rotations $(\alpha^A, \beta^A, \gamma^A, \alpha^B, \beta^B)$ about the z- and y-axes.*

LEMMA 4.2.

$$C(\mathbf{R}) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell}^{\ell} (-1)^m (-1)^{m'} D_\ell^{m,m'}(\mathbf{R}) \int_{\mathbb{R}^+} \hat{a}_{\ell -m}(r) \overline{\hat{b}_{\ell -m'}}(r) r^2 dr. \qquad (4.2)$$

To derive the expression for general rigid-body correlations (Figure 4.1) $C(\mathbf{R}, \mathbf{t}) = \int_{\mathbb{R}^3} A(\mathbf{x}) \overline{B(\mathbf{R}\mathbf{x} + \mathbf{t})} d\mathbf{x}$, we can use Equation 3.10 along with an elementary fact: every rigid-body motion $(\mathbf{R}, \mathbf{t})$ can be factored into a combination of five rotations and a single translation about the z-axis[3]. Let these five rotations be parametrized by z-y-z Euler angles $\mathbf{R}^A = (\alpha^A, \beta^A, \gamma^A)$ and $\mathbf{R}^B = (0, \beta^B, \gamma^B)$. Then we obtain, for the mixed-basis functions:

LEMMA 4.3.

$$C(\mathbf{R}, \mathbf{t}) = \sum_{klmn} \hat{a}_{k\ell m} D_\ell^{n,m}(\mathbf{R}^A) \sum_{k'\ell' m'} (-1)^n \hat{b}_{k'\ell' m'} D_{\ell'}^{-n,m'}(\mathbf{R}^B) T_{k\ell,k'\ell'}^{|n|}(z). \qquad (4.3)$$

Following an observation in Garçon et. al [11], it is not as efficient to use the pure spherical basis expansions to express a general rigid-body correlation. Instead, Equation 4.2 is used along with a scan of the translational degrees of freedom, in which the basis coefficients are recomputed for each distinct $\mathbf{t} \in \mathbb{R}^3$.

Lemmas 4.2 and 4.3 have also been obtained in Kovacs and Wriggers [16] and Ritchie [30]. To our knowledge, this is the first explicit establishment of the straightforward Lemma 4.1.

5. The tools: Wigner-d-Chebyshev, Chebyshev-exponential and $SO(3)$ transforms. The Wigner-d polynomials are the main obstacle between Equations 4.1 and 4.3 and their expression as exponential, FFT-amenable sums; existing approaches to computing rigid-body correlations use one of a pair of well-known transformations

[3]It is enough to see that every translation $\mathbf{t}$ can be expressed as two rotations and a single translation along the z-axis. Starting at the origin, the point $\mathbf{t}$ can be reached by translating along the z-axis by $\|\mathbf{t}\|$, and then rotating about the z and $y-$axes by θ and ϕ, the spherical coordinates of $\mathbf{t}$.

to express these polynomials as exponentials. Here we use a combination of Wigner-d-Chebyshev/Chebyshev-exponential transforms [27], briefly described below, to effect the same conversion. This conversion enables us to compute Equations 4.1 and 4.3 over arbitrary samples in $\mathbb{R}^3 \times SO(3)$.

Consider the summation

$$\sum_{\ell=\max|m|,|n|} \hat{f}_{\ell mn} d_\ell^{m,n}(\cos\beta).$$

Potts et.al [27] convert the above summation to an exponential sum via a two step process. The first step is the Wigner-d-Chebyshev transform. Let $T_\ell(\cos\beta)$ be the family of Chebyshev polynomials of the first kind.

DEFINITION 5.1. ***Wigner-d-Chebyshev Transform.*** *Let $\hat{f}_{\ell mn} \in \mathbb{C}$. The Wigner-d-Chebyshev transform of $\hat{f}_{\ell mn}$ is the set of coefficients $\hat{g}_{\ell mn} \in \mathbb{C}$ such that*

$$\sum_{\ell=\max|m|,|n|}^{L} \hat{f}_{\ell mn} d_\ell^{m,n}(\cos\beta) = \begin{cases} \sum_{\ell=0}^{L} \hat{g}_{lmn} T_\ell(\cos\beta) & \text{if } m+n \text{ is even} \\ \sin\beta \sum_{\ell=0}^{L-1} \hat{g}_{lmn} T_\ell(\cos\beta) & \text{if } m+n \text{ is odd.} \end{cases} \tag{5.1}$$

The Wigner-d-Chebyshev transform is computed by using a three-term recurrence relationship for the Wigner-d functions, along with a fast polynomial transform [28]. This transform is independent of β, and can be performed in $\mathcal{O}(L\log^2 L)$ steps for each distinct m, n.

The second step, the Chebyshev-exponential transform, involves converting Chebyshev coefficients $\hat{g}_{\ell mn}$ into exponential coefficients $\hat{h}_{\ell mn}$.

DEFINITION 5.2. ***Chebyshev-exponential Transform.*** *Let $\hat{g}_{\ell mn} \in \mathbb{C}$. The Chebyshev-exponential transform of $\hat{g}_{\ell mn} \in \mathbb{C}$ is the set of coefficients $\hat{h}_{\ell mn} \in \mathbb{C}$ such that*

$$\sum_{\ell=0}^{L} \hat{g}_{\ell mn} T_\ell(\cos\beta) = \sum_{\ell=-L}^{L} \hat{h}_{\ell mn} \exp(-i\ell\beta). \tag{5.2}$$

The latter is a straightforward computation, taking advantage of the relationship $\cos\beta = \frac{\exp(i\beta)+\exp(i\beta)}{2}$. Like the Wigner-d-Chebyshev transform, it is independent of β, and takes $\mathcal{O}(L)$ steps. The transformation from Wigner-d coefficients $\hat{f}$ to exponential coefficients $\hat{h}$ thus takes a total of $\mathcal{O}(L\log^2 L)$ steps. This pair of conversions is at the heart of the fast SO(3) transform.

5.1. The $SO(3)$ Fourier transform. To efficiently calculate either of Equations 4.1 and 4.3, we use the fast SO(3) Fourier Transform [27].

DEFINITION 5.3. ***SO*(3) *Fourier transform.*** *Let $\hat{f}_{\ell mn} \in \mathbb{C}$, and $(\alpha_q, \beta_q, \gamma_q) \in SO(3), q \in \{1\ldots, Q\}$. The SO(3) Fourier transform of $\hat{f}_{\ell mn}$ is the set of coefficients $F(\alpha_q, \beta_q, \gamma_q) \in SO(3)$ such that*

$$F(\alpha_q, \beta_q, \gamma_q) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \sum_{n=-\ell}^{\ell} \hat{f}_{\ell mn} \tilde{D}_\ell^{m,n}(\alpha_q, \beta_q, \gamma_q). \tag{5.3}$$

This sum can be transformed into an FFT-amenable sum [27]. We have

$$F(\alpha_q, \beta_q, \gamma_q) = \sum_{m=-L}^{L} \sum_{n=-L}^{L} \sum_{\ell=\max(|m|,|n|)}^{L} \hat{f}_{\ell mn}$$

$$\times \exp(-im\alpha_q) d_\ell^{m,n}(\cos\beta_q) \exp(-in\gamma_q).$$

Using the Wigner-d-Chebyshev and Chebyshev-exponential transforms, the innermost sum can be transformed into a triple exponential sum:

$$F(\alpha_q, \beta_q, \gamma_q) = \sum_{m=-L}^{L} \sum_{n=-L}^{L} \sum_{\ell=-L}^{L} \hat{h}_{\ell mn} \exp(-im\alpha_q) \exp(-i\ell\beta_q) \exp(-in\gamma_q). \quad (5.4)$$

If the nodes $(\alpha_q, \beta_q, \gamma_q)$ are equispaced, an FFT can be used to compute Equation 5.4. If they are non-equispaced, however, the non-uniform fast Fourier transform [15] comes to our rescue.

DEFINITION 5.4. ***Non-equispaced Fast Fourier Transform.*** *Let $\hat{h}_{\ell mn} \in \mathbb{C}$. Then the non-equispaced Fast Fourier Transform (NFFT) computes Equation 5.4 for arbitrary nodes $i \in \{1 \dots Q\}$ in $\mathcal{O}(L^3 \log L + Q)$ steps.*

An explanation of the techniques behind the NFFT is beyond the scope of this work; however, in the discussions to follow, we shall see how the non-equispaced nature of the NFFT actually provides PF*corr* with a crucial advantage.

We now have a set of techniques to compute the SO(3) Fourier transform introduced at the beginning of this section.

LEMMA 5.5. *The SO(3) Fourier transform can be computed in $\mathcal{O}(L^3 \log^2 L + Q)$ steps (Potts et.al [27]).*

Proof. Use the Wigner-d-Chebyshev and Chebyshev-exponential transforms to convert Equation 5.3 to Equation 5.4 in $\mathcal{O}(L^3 \log^2 L)$ steps, and then use the NFFT to compute the remaining sum in $\mathcal{O}(L^3 \log L + Q)$ steps. □

6. Rigid-body correlations: main results. We use the machinery of the non-uniform SO(3) transform to compute Equations 4.1, 4.2, and 4.3 (Theorems 6.1, 6.2, and 6.3). We also outline a way to speed up computations of the SO(3) translation matrix entries (Theorem 6.4).

THEOREM 6.1. *The pure rotational correlation $C(\mathbf{R})$ (Equation 4.1) can be computed in $\mathcal{O}(L^4 + N_{\mathbf{R}})$ steps, where $N_{\mathbf{R}}$ is the number of distinct rotations $\mathbf{R}$.*

THEOREM 6.2. *The pure rotational correlation $C(\mathbf{R})$ (Equation 4.2) can be computed in $\mathcal{O}(L^3 \log^2 L + N_{\mathbf{R}} + L^3 I)$ steps, where $N_{\mathbf{R}}$ is the number of distinct rotations $\mathbf{R}$ and I is the complexity of computing the integral $\int_{R^+} \hat{a}(r)\hat{b}(r)r^2 dr$ for a given pair of scalar-valued functions $\hat{a}, \hat{b} : \mathbb{R}^+ \mapsto \mathbb{C}$.*

THEOREM 6.3. *The general rigid-body correlation $C(\mathbf{R}, \mathbf{t})$ (Equation 4.3) can be computed in $\mathcal{O}(L^6 + L^4 N_{\mathbf{R}^B} + N_{\mathbf{R}^B} N_{\mathbf{R}^A}) N_t$ steps, where $N_{\mathbf{R}^A}$ and $N_{\mathbf{R}^B}$ are the number of rotations of A and B respectively, and N_t is the number of 1 dimensional translations.*

THEOREM 6.4. *The translation matrix entries in Equation 3.5 for $SO(3)$ can be computed in $\mathcal{O}(L^7 + L^6 N_t)$, where N_t is the number of 1 dimensional translations.*

Theorems 6.1 and 6.3 are the main correlation-based results of this work, and are respectively proven by the pair of algorithmic Recipes 1 and 2 presented in the appendix. Theorem 6.2 follows directly from Theorem 6.1.

With these theorems established, we can now outline algorithms to perform fast rigid-body correlations given a pair of scalar-valued functions as input. Algorithm 1 uses the mixed radial/spherical basis, while Algorithm 2 uses the pure spherical har-

monic basis functions.

Algorithm 1: Fast Rotational Matching with mixed radial/spherical basis functions

Input: L : Expansion degree;
G : Spherical grid with sizes N_r, N_θ, N_ϕ in the radial, polar and azimuthal directions respectively. Let $N = \max(N_r, N_\theta, N_\phi)$;
$A, B : \mathbb{R}^3 \mapsto \mathbb{C}$: scalar-valued functions sampled on G centered at $r = 0$;
$\mathcal{M} \subset \mathbb{R}^3 \times SO(3)$: a finite set of rigid-body motions;

foreach (k, ℓ, m) *with* $|m| \leq \ell \leq k \leq L$ **do**
  Calculate the coefficients $\hat{a}_{k\ell m}$ and $\hat{b}_{k\ell m}$ using Equation 3.5;
end
if $\mathbf{t} == \mathbf{0}$ $\forall (\mathbf{R}, \mathbf{t}) \in \mathcal{M}$ **then**
Find the maximum value of $C(\mathbf{R}) = \int_{\mathbb{R}^3} A(\mathbf{x}) B(\mathbf{R}\mathbf{x}) d\mathbf{x}$ $\forall \mathbf{R} \in \mathcal{M}$ using the steps in the proof of Theorem 6.1;
else Find the maximum value of $C(\mathbf{R}, \mathbf{t}) = \int_{\mathbb{R}^3} A(\mathbf{x}) B(\mathbf{R}\mathbf{x} + \mathbf{t}) d\mathbf{x}$ $\forall (\mathbf{R}, \mathbf{t}) \in \mathcal{M}$ using the steps in the proof of Theorem 6.3;

Output: The maximum correlation $C \in \mathbb{C}$ between A and B;
Complexity: $\mathcal{O}(\mathcal{C}_{\text{coeff}} + \mathcal{C}_{\text{PF}corr})$ flops, where $\mathcal{C}_{\text{coeff}} = \mathcal{O}(L^3 N^3)$ is the complexity of computing the coefficients $\hat{a}_{klm}$, and $\mathcal{C}_{\text{PF}corr} = \mathcal{O}(L^4 + N_{\mathbf{R}})$ in the pure rotational case or $\mathcal{O}(L^6 + L^4 N_{\mathbf{R}^B} + N_{\mathbf{R}^B} N_{\mathbf{R}^A}) N_t$ in the general case;

Algorithm 2: Fast Rotational Matching with pure spherical harmonic basis functions

Input: L : Expansion degree;
G : Spherical grid with sizes N_r, N_θ, N_ϕ in the radial, polar and azimuthal directions respectively. Let $N = \max(N_r, N_\theta, N_\phi)$;
$A, B : \mathbb{R}^3 \mapsto \mathbb{C}$: scalar-valued functions sampled on G centered at $r = 0$;
$\mathcal{T} \subset \mathbb{R}^3 \times SO(3)$: a finite set of pairs $\{(\mathbf{t}, \mathcal{R})\}$, where $\mathbf{t} \in \mathbb{R}^3$ is a translation and $\mathcal{R} \subset SO(3)$ is a finite set of rotations corresponding to $\mathbf{t}$;

foreach $r \in G$ **do**
  foreach (ℓ, m) *with* $|m| \leq \ell \leq L$ **do**
    Compute $\hat{a}_{lm}(r)$ using Equation 3.7;
  end
end
foreach $(\mathbf{t}, \mathcal{R}) \in \mathcal{T}$ **do**
  Translate $B(\mathbf{x})$ by $\mathbf{t}$;
  foreach (ℓ, m) *with* $|m| \leq \ell \leq L$ **do**
    Compute $\hat{b}_{lm}(r)$ using Equation 3.7;
  end
  Compute $C(\mathbf{R}) = \int_{\mathbb{R}^3} A(\mathbf{x}) B(\mathbf{R}(\mathbf{x} + \mathbf{t})) d\mathbf{x}$ $\forall \mathbf{R} \in \mathcal{R}$ using the steps in the proof of Theorem 6.2.
end

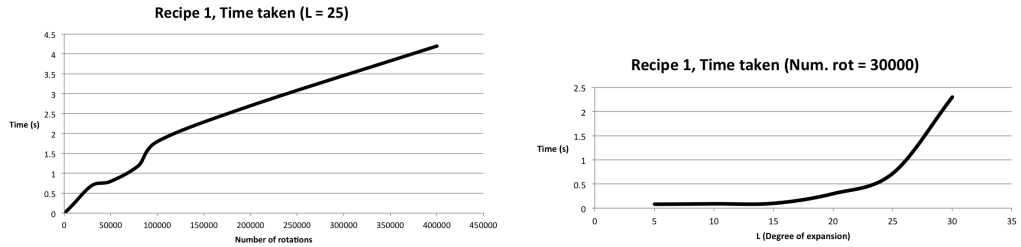
Output: The maximum correlation $C \in \mathbb{C}$ between A and B;
Complexity: $\mathcal{O}((\mathcal{C}_{\text{coeff}} + \mathcal{C}_{\text{PF}corr})|\mathcal{T}|)$ flops, where $\mathcal{C}_{\text{coeff}} = \mathcal{O}(N^2 L^2)$, and $\mathcal{C}_{\text{PF}corr} = \mathcal{O}(L^3 \log L + N_{\mathbf{R}})$;

7. Rigid-body correlations: numerical results and discussion. There are three sources of error in PF*corr*. The first is the expansion error, i.e., the error

L	A/M/MC*	A/M/MC†
3	1.21e-5/1.48e-5/1.56e-10	1.51e-4 / 1.82e-4 / 1.81e-10
4	1.93e-5/1.22e-4/1.71e-10	1.26e-4 / 1.75e-4/1.84e-10
5	1.91e-5/1.23e-4/1.83e-10	1.38e-4 / 1.68e-4/ 1.85e-10
6	1.37e-5/1.67e-5/1.53e-10	1.44e-4 / 1.88e-4 / 1.76e-10
7	1.58e-5/1.71e-5 /1.78e-10	1.32e-4 / 1.51e-4 / 1.81e-10
8	1.61e-5 /1.78e-5/1.72e-10	1.22e-4 / 1.48e-4/ 1.71e-10
9	1.67e-5/1.70e-5/1.72e-10	1.31e-4 / 1.44e-4 / 1.82e-10
10	1.61e-5/1.9e-5 /1.77e-10	1.34e-4/ 1.39e-4 / 1.85e-10
11	1.73e-5/ 1.88e-5/1.79e-10	1.32e-4/ 1.38e-4/ 1.84e-10
12	1.72e-5/ 1.82e-5 /1.41e-10	1.32e-4 / 1.4e-4 / 1.81e-10

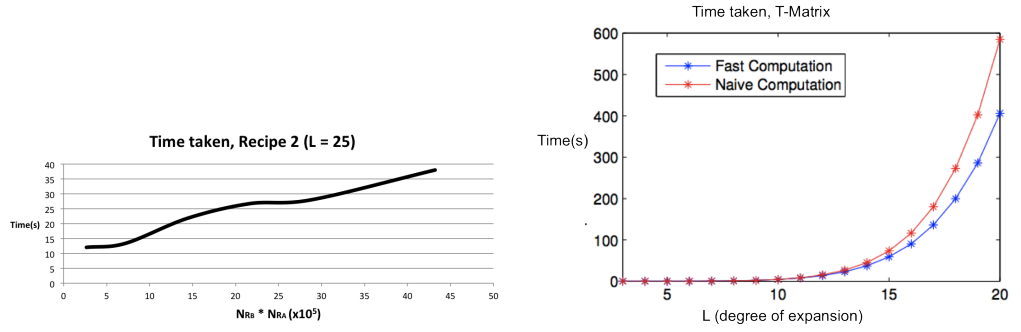
TABLE 7.1

Errors between the naively computed correlation and the correlation as computed by Recipes 1 *and* 2 *for two real-valued functions* $A, B : \mathbb{R}^3 \to \mathbb{R}$ *at varying expansion degrees over* 500 *randomly-generated rigid-body rotations. In each case, error* $= \frac{|naive - recipe|}{|naive|}$ *.* * *: Average error/maximum error/maximum complex value for Recipe* 1*.* † *: Average error/maximum error/maximum complex value for Recipe* 2*. Beyond* $L = 12$*, the naive correlation is exceedingly slow to compute. The above experiment was conducted with radial/spherical bases; similar results hold for pure spherical bases, as the speedup scheme for these bases is the same as that used for Recipe* 1*.*



(A) (B)

FIG. 7.1. *Time taken by each of the algorithms in Section 6. (A) Time taken by Recipe* 1 *at a fixed degree. (B) Time taken by Recipe* 1 *at a fixed number of rotations. (C). Time taken by Recipe* 2 *at fixed degree. (D) Time taken by the T-Matrix algorithm. See also Figure 7.2.*



(A) (B)

FIG. 7.2. *Time taken by each of the algorithms in Section 6, part* 2*. (A) Time taken by Recipe* 2 *at fixed degree. (B) Time taken by the T-Matrix algorithm. See also Figure 7.1.*

induced by truncating the basis expansion at a finite value of L. The second is the representation error, i.e., the error induced in numerically integrating the coefficients in Equation 3.5. The third is the NFFT error, i.e., the error induced by approximating the exponential sums by the NFFT.

Following Ritchie, Ritchie et. al, and Garçon et. al [11, 30, 32], the first two sources of error can be respectively mitigated by choosing an expansion degree between $20 \leq L \leq 25$, and using a single-point quadrature rule. We provide further evidence supporting the former assertion in the following subsection.

The NFFT approximates exponential sums with a kernel basis expansion, providing a choice of several kernels, and several parameters govern the actual error of the expansion. In our implementation, we choose the Gaussian kernel with an oversampling factor of 3 (See Potts et. al [29]), resulting in the errors in Table 7.1. We note that, in solutions to the correlation problem, the absolute value of a correlation is less important than its value relative to other rigid-body rotations, i.e., the ability of the search scheme to discriminate between two different rigid-body motions. A measure of this ability is presented in the following subsection in the context of sampling arbitrary subsets of $SE(3)$.

We provide timing information in Figure 7.1 and 7.2[4]. For Recipe 1 (Figure 7.1), we see that the linear scaling with respect to the number of rotations and the quartic scaling with respect to expansion degree predicted by Theorem 6.1 are reproduced by the implementation. For Recipe 2 (Figure 7.2), the scaling with respect to expansion degree is not very important, as typically L^6, the leading expansion term, is much less than $L^4 N_{R_B}$, which, for practical correlation problems, is in turn less than the product of the number of rotations $N_{R_A} N_{R_B}$. We hence examine how Recipe 2 scales with respect to the product $N_{R_A} N_{R_B}$; Figure 7.2(A), shows that the scaling is linear, as expected.

For the T-Matrix computation in Theorem 6.4 (Figure 7.2 (B)), a dramatic speedup with respect to the naive algorithm is observed in $L \geq 10$ regime, where the L^7 v/s L^8 scaling is apparent. However, for typical values of L (see following paragraph), the computation times are still too slow to be usable in the inner loop of any Fourier-based correlation approach, including our own. Like prior work that uses the T-Matrix (see the introduction for an overview), we thus prefer to precompute and store T-Matrix entries for given values of z and λ (See Definition 3.5).

From a practical standpoint, our rigid-body correlation search is seen to be a viable, if somewhat slower, alternative to existing rigid-body correlation search techniques. Most of the degradation in performance is due to the NFFT, which uses, in its implementation, an oversampled FFT to enable the non-uniformity inherent to it. Following Ritchie [32], we choose L to typically lie between 20 and 25, in which case typical run times for an exhaustive correlation involving about $1.5e7$ distinct rigid-body samples lie between 2 and 3 minutes. We also note that, other than the argument in Section 7.1, there is no reason to prefer the non-uniformity inherent to PF*corr*, and, if performance is a concern, each of the steps involving the NFFT can be replaced by the equispaced FFT.

7.1. Sampling arbitrary subsets of $SE(3)$; addressing the drawbacks of existing techniques. The main advantage of PF*corr* is in sampling arbitrary (finite) subsets of the space $SE(3) = \mathbb{R}^3 \times SO(3)$ of rigid-body motions. In our implementa-

[4]All timing information is from a single-threaded, dual core Macbook Pro at 2.5 GhZ with 8GB of RAM.

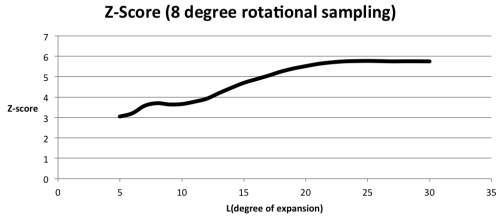


FIG. 7.3. *Average top-ranking Z-Scores for recipe* 1 *at varying degrees. The rotational sampling fineness is fixed at* 8°.

tion, this is as simple as specifying a set of rigid-body motions on which correlations are to be performed. By contrast, all prior techniques *require* an equispaced Euler Angular grid for rotational search, a property that results in a highly non-uniform search of the space of rotations (See Drawback 2 in the introduction).

What subset of $\mathbb{R}^3 \times SO(3)$ is best? For exhaustive correlations between a pair of scalar-valued functions, one good answer to this question is: one that *uniformly* samples the space of rotations $SO(3)$. As we mention in the introduction, most of the uncertainty in the rigid-body correlation problem lies in the space of rigid-body rotations, and it is thus more important to sample this space as uniformly as possible. There are several existing techniques that, given an angular sampling criterion, provide a set of samples that are uniform with respect to accepted metrics of uniformity. We use the approach in Mitchell [22], in which the metrics of *local separation* and *global coverage* compete to provide a set of highly uniform samples in $SO(3)$. See also Figure 2.1.

The ability to sample and correlate over arbitrary subsets of $SE(3)$ is only useful if, for a particular expansion degree, the fineness of the rotational sample size does not exceed the accuracy with which $\hat{a}_{klm}$ and $\hat{b}_{klm}$ represent A and B respectively (See Equation 3.6). Such a scenario would give rise to correlations that are so close to each other as to be essentially indistinguishable, and would result in a set of correlations clustered around the average. To measure this tendency, we compute the z-score $z = \frac{x-\mu}{\sigma}$, a measure of the distance of each individual correlation from the average. The results, in Figure 7.3, indicate that (A) the top-ranking Z-score increases with increase in degree, as expected, leveling off at $L \geq 20$, where the error due to floating-point calculations begins to rival the error due to representation, and (B) even at very low expansion degrees, the top-ranking score is 3 standard deviations from the mean, indicating a very high confidence. Figure 7.3 also presents another argument as to why the regime $20 \leq L \leq 25$ is best, as the latter provides a balance between the errors of representation and floating-point computation.

8. Flexible correlations: main results. We present an algorithm (Algorithm 3) for domain-based protein matching. This algorithm, given as input

1. A protein $\mathcal{P}$,
2. A hierarchical domain decomposition, defined in Section 8.1, of $\mathcal{P}$,
3. A scalar-valued function $B : \mathbb{R}^3 \mapsto \mathbb{R}$ representing a stationary target, and,
4. A scalar-valued representation A of $\mathcal{P}$,

produces as output the optimal correlation between A and B under rigid-body motions of the domains of $\mathcal{P}$. Algorithm 3 makes use of the ability of PF*corr* to

uniformly sample arbitrary subsets of $\mathbb{R}^3 \times SO(3)$.

Algorithm 3: Greedy multi-domain matching

Input:

1. $\mathcal{P}$: Protein;
2. $\mathcal{DD} = \{D_i, MG\}, i \in \{1 \dots N_D\}$: A domain decomposition of $\mathcal{P}$;
3. $\mathcal{R}(\mathcal{DD}_i)$:
   An operator that converts $D_i \in \mathcal{DD}$ into a representation $A_i : \mathbb{R}^3 \mapsto \mathbb{R}$;
4. $A : \mathbb{R}^3 \mapsto \mathbb{R}$: Scalar-valued function representing $\mathcal{P}$, computed using $\mathcal{R}$;
5. $B : \mathbb{R}^3 \mapsto \mathbb{R}$: Target scalar-valued function;
6. PQ :
   Empty priority queue with elements (j, r), $j \in \mathbb{Z}^+, r \in \mathbb{R}$ ordered least-first w.r.t r;

Output: The optimal correlation between A_i and B under rigid-body transformations of A_i, $i \in \{1 \dots N_D\}$.

Use PF*corr* to find the optimal rigid-body transformation $(\mathbf{R}, \mathbf{t})$ relating A to B;
foreach $D_i \in \mathcal{DD}$ **do**
    Compute the correlation $C_i \leftarrow \int_{\mathbb{R}^3} A_i B d\mathbf{x}$ between each domain D_i and the target B;
    Push (i, C_i) to PQ;
end
$i \leftarrow 1$;
while $i \leq N_D$ **do**
    $k \leftarrow PQ[N_D - i - 1].j$;
    $D_i \leftarrow D_k$;
    $i \leftarrow i + 1$;
end
foreach $D_i \in \mathcal{DD}$, $i \neq 1$ **do**
    Using flexors $F_{i-1,i}$, compute the set of relative motions $T_{i-1,i} \leftarrow \{(\mathbf{R}^k_{i-1,i}, \mathbf{t}^k_{i-1,i})\}$, $k \in \{1 \dots N^i_{\mathbf{T}}\}$ of D_i relative to D_{i-1};
    Compute the set of absolute motions $T_i \leftarrow \{(\mathbf{R}^k_i, \mathbf{t}^k_i)\}$, $k \in \{1 \dots N^i_{\mathbf{T}}\}$ for each rigid-body transformation in the set $T_{i-1,i}$ relative to the stationary domain D_1;
end
foreach $(i, C_i) \in PQ$ **do**
    Use PF*corr* to find the optimal rigid-body transformation $(\mathbf{R}_i, \mathbf{t}_i) \in T_i$ relating A_i to B;
end

Complexity: $\mathcal{O}(C_{\mathrm{PF}corr} N_D)$ flops, where $C_{\mathrm{PF}corr}$ is the complexity of PF*corr*.

8.1. Domain-based protein flexibility framework. We assume a generic framework for domain-based protein flexibility. This framework consists of ideas from domain-decomposition of proteins that have existed in various forms over the past decade (see especially Maiorov and Abagyan [20]), as well as a set of techniques, described, for instance, in Bajaj et. al [3], to assign motions to each of these domains.

Let a protein crystal structure $\mathcal{P}$ comprise a set of atoms. Designate a subset of $\mathcal{P}$

as a domain D. A domain decomposition of $\mathcal{P}$ is a set $\mathcal{DD} = \{D_i\}, 1 \leq i \leq n_{\mathcal{DD}}$, where D_i is a domain. A hierarchical domain decomposition $\mathcal{HD} = \{\mathcal{DD}_i\}, 1 \leq i \leq n_{\mathcal{HD}}$ is a set of domain decompositions $\mathcal{DD}_i$ such that each domain in $\mathcal{DD}_i$ is a subdomain of some domain in $\mathcal{DD}_{i-1}$ (See, for example, Bettadapura et. al [4]). For each $\mathcal{DD}_i$ of the hierarchical domain decomposition $\mathcal{HD}$, a motion graph MG specifying relative motions between domains of $\mathcal{DD}_i$ can be specified. The motion graph consists of a set of edges F_{ij}, called flexors, between pairs of domains i, j that undergo relative motion. The geometric properties of each flexor imply a set of rigid-body transformations $(\mathbf{R}^k_{i,j}, \mathbf{t}^k_{i,j})$, $k \in \{1 \dots N_{\mathbf{T}}\}$ applied to D_j relative to D_i [3].

8.2. Algorithm for flexible matching. Algorithm 3 applies to a particular domain decomposition of $\mathcal{P}$, i.e, it applies to a particular index in the hierarchical domain decomposition of $\mathcal{P}$. It uses the ability of PF*corr* to sample arbitrary subsets of $SE(3)$ to match representations of domains $A_i \in A$ to a target scalar-valued function $B : \mathbb{R}^3 \mapsto \mathbb{R}$. Note by contrast that a classic equispaced Fourier-based correlation scheme would not be able to perform the tasks in Algorithm 3 without also producing several results that do not belong to the chosen subset of $SE(3)$. This focusing property enables PF*corr* to combine the merits of both local and global optimization schemes in the following sense. The algorithm is *local* in that it is restricted to the chosen subset of $SE(3)$, but *global* in that it samples that subset exhaustively. It thus combines the speed of a local search without being sensitive, as local search algorithms are, to local optima.

9. Conclusion. We have presented PF*corr*, a non-uniform correlation search scheme. PF*corr* displays the following properties: (A) It is sampling robust, making searches over arbitrary subsets of $SE(3)$ efficient while retaining the capabilities of classical exhaustive Fourier-based search schemes, (B) It is compatible with existing equispaced FFT-based techniques, in the sense that its non-equispaced nature is desirable but not necessary, and (C) Its algorithms extend to the rotationally exhaustive paradigms in Kovacs and Wriggers, Ritchie, and Ritchie et. al [11, 17, 30, 32]. We have also presented an algorithm to compute translation matrix entries for $SO(3)$ that achieves a better scaling than existing naive algorithms. Finally, we have presented an algorithm for multi-dimensional flexible correlations that leverages the sampling robustness of PF*corr*. PF*corr* applies to several fields within computational biology, including, most notably, molecular fitting and docking, where the above properties make it a natural and efficient tool for correlation-amenable search.

Acknowledgment. This work was supported in part by grants from the NIH R01 GM074258, R01-GM073087,R01-EB004873. Much of the work on this paper was accomplished when Benedikt Bauer and Antje Vollrath were visiting the University of Texas at Austin. Our in-house molecular modeling and visualization software tool, called VolRover, was used as the visualization front end for docking. The VolRover program is in the public domain and can be freely downloaded from our center's software website (`http://www.ices.utexas.edu/CVC/software/`).

Appendix. *Proof of Lemma 3.3.* Let $\lambda = 1$. We have

$$\int_0^\infty R^\ell_k(r) R^\ell_n(r) r^2 \mathrm{d}r = \beta_{k\ell}\beta_{n\ell} \int_0^\infty \mathrm{e}^{-r^2} r^{2\ell} L^{\ell+\frac{1}{2}}_{k-\ell-1}\left(r^2\right) L^{\ell+\frac{1}{2}}_{n-\ell-1}\left(r^2\right) r^2 \mathrm{d}r.$$

By substituting first $r^2 = x$, and then $\alpha = \ell + \frac{1}{2}, k' = k - \ell - 1$ and $n' = n - \ell - 1$, we get

$$\begin{aligned}\int_0^\infty R_k^\ell(r) R_n^\ell(r) r^2 \mathrm{d}r &= \beta_{k\ell}\beta_{n\ell} \int_0^\infty \mathrm{e}^{-x} x^\ell L_{k-\ell-1}^{\ell+\frac{1}{2}}(x) L_{n-\ell-1}^{\ell+\frac{1}{2}}(x) \frac{x}{2\sqrt{x}} \mathrm{d}x \\ &= \frac{1}{2}\beta_{k\ell}\beta_{n\ell} \int_0^\infty \mathrm{e}^{-x} x^\alpha L_{k'}^\alpha(x) L_{n'}^\alpha(x) \mathrm{d}x.\end{aligned}$$

Using the orthogonality of the Laguerre polynomials, and back-substituting k' and n', this simplifies to

$$\begin{aligned}\int_0^\infty R_k^\ell(r) R_n^\ell(r) r^2 \mathrm{d}r &= \frac{1}{2}\beta_{k\ell}\beta_{n\ell} \frac{\Gamma(k' + \alpha + 1)}{k'!} \delta_{k'n'} = \frac{1}{2}\beta_{k\ell}\beta_{n\ell} \frac{\Gamma(k + \frac{1}{2})}{(k-\ell-1)!} \delta_{kn} \\ &= \beta_{k\ell}^2 \frac{\Gamma(k + \frac{1}{2})}{2(k-\ell-1)!} \delta_{kn} = \delta_{kn}\end{aligned}$$

after inserting $\beta_{k\ell}$. The extension to general λ is straightforward. □

Proof of Lemma 4.1.

Using the basis expansions of A and B,

$$C(\mathbf{R}) = \int_{\mathbb{R}\times\mathbb{S}^2} \sum_{k\ell m} \hat{a}_{k\ell m} R_k^\ell(r) Y_\ell^m(\mathbf{u}) \sum_{k'\ell'm'm''} \overline{\hat{b}_{k'\ell'm'}} R_{k'}^{l'}(r) \overline{D_\ell^{m'',m'}(\mathbf{R}) Y_{\ell'}^{m''}(\mathbf{u})} r^2 dr d\mathbf{u}. \tag{9.1}$$

Putting the basis-function orthogonality conditions

$$\int_{\mathbb{R}} R_k^\ell(r) R_{k'}^{\ell'}(r) r^2 dr = \delta_{kk'} \tag{9.2}$$

and

$$\int_{\mathbb{S}^2} Y_\ell^m(\mathbf{u}) \overline{Y_{\ell'}^{m'}(\mathbf{u})} d\mathbf{u} = \delta_{mm'}\delta_{\ell\ell'} \tag{9.3}$$

into Equation 9.1 yields the desired result. □

Proof of Lemma 4.3.

$$C(\mathbf{R}, \mathbf{t}) = \int_{\mathbb{R}^3} A(\mathbf{x}) \overline{B(\mathbf{R}\mathbf{x} + \mathbf{t})}$$

$$\Rightarrow C(\alpha^A, \beta^A, \gamma^A, \Delta z, \beta^B, \gamma^B) = \int_{\mathbb{R}^+\times\mathbb{S}^2} \sum_{klmn} \hat{a}_{klm} D_\ell^{n,m}(\mathbf{R}^A) R_k^\ell(r) Y_\ell^n(\mathbf{u})$$

$$\times \sum_{k'\ell'm'n'} \overline{\hat{b}_{k'\ell'm'} D_{\ell'}^{n'm'}(\mathbf{R}^B)} \sum_{j=0}^{\infty} \sum_{h=0}^{j-1} T_{jh,k\ell}^{|n'|}(-z) R_j^h(r) \overline{Y_h^{n'}(\mathbf{u})} r^2 d\mathbf{u} dr,$$

which, after a routine invocation of the orthogonality conditions in Equations 9.2 and 9.3, reduces to the desired result. □

Proof of Theorem 6.1.
Rewriting Equation 4.1 as

$$C(\mathbf{R}) = \sum_{\ell=0}^{L-1} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell}^{\ell} \sum_{k=\ell+1}^{L} (-1)^m \hat{a}_{k\ell-m} (-1)^{m'} \times \overline{\hat{b}_{k\ell-m'}} D_\ell^{m,m'}(\alpha,\beta,\gamma)$$

enables us to compute it in two steps.
Recipe 1.

1. Compute the innermost sum

$$\hat{f}_{\ell m m'} = \sum_k (-1)^{m'} \hat{a}_{k\ell-m} \overline{\hat{b}_{k\ell-m'}}$$

   in $\mathcal{O}(L^4)$ steps.
2. Use the SO(3) Fourier transform to compute the remaining sum

$$C(\mathbf{R}) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \sum_{n=-\ell}^{\ell} \hat{f}_{\ell m n} \exp(-im\alpha) d_\ell^{m,n}(\cos\beta) \exp(-in\gamma)$$

   in $\mathcal{O}(L^3 \log^2 L + N_{\mathbf{R}})$ steps, where $N_{\mathbf{R}}$ is the number of unique rotations $\mathbf{R}$.

The overall cost is $\mathcal{O}(L^4)$, the cost of the most expensive first step. □

Proof of Theorem 6.2. There are $\mathcal{O}(L^3)$ integrals $\int_{\mathbb{R}^+} \hat{a}(r)\hat{b}(r)r^2 dr$, and once these have been computed, the triple sum in Equation 4.2 is an $SO(3)$ Fourier transform, and can be computed, from Lemma 5.5, in $\mathcal{O}(L^3 \log^2 L + Q)$ steps. □

Proof of Theorem 6.3.
Rewriting Equation 4.3 as

$$C = \sum_{l=0}^{L-1} \sum_{m=-\ell}^{\ell} \sum_{n=-\ell}^{\ell} \sum_{k=\ell+1}^{L} \hat{a}_{k\ell m} D_\ell^{n,m}(\mathbf{R}^A)$$

$$\times \sum_{m'=-(L-1)}^{L-1} \sum_{\ell=\max(|m'|,|n|)}^{L-1} (-1)^n D_{\ell'}^{-n,m'}(\mathbf{R}^B) \times \sum_{k'=\ell'+1}^{L} \hat{b}_{k'\ell'm'} T_{k\ell,k'\ell'}^{|n|}(-z)$$

enables us to compute it in four steps.
Recipe 2.

1. Compute

$$\hat{c}_{k\ell\ell' nm'} = \sum_{k'=\ell'+1}^{L} \hat{b}_{k'\ell'm'} T_{k\ell,k'\ell'}^{|n|}(-z)$$

   in $\mathcal{O}(L^6)$ steps.
2. Via the Wigner-d-Chebyshev and Chebyshev-exponential transforms, convert

$$\sum_{m'=-(L-1)}^{L-1} \sum_{\ell'=\max(|m'|,|n|)}^{L-1} \hat{c}_{k\ell\ell' nm'} D_{\ell'}^{-n,m'}(\mathbf{R}^B)$$

to

$$\sum_{m'=-L+1}^{L-1} \sum_{\ell'=-L+1}^{L-1} \hat{d}_{kll'nm'} e^{-im'\gamma^B} e^{-i\ell'\beta^B}$$

in $\mathcal{O}(L^5)$ steps.

3. Using the NFFT, compute

$$\hat{e}_{kln}(-\Delta z, \beta^B, \gamma^B) = \sum_{m'=-L+1}^{L-1} \sum_{\ell'=-L+1}^{L-1} \hat{d}_{k\ell\ell'nm'} \exp(-im'\gamma^B) \exp(-i\ell'\beta^B)$$

for all sets of (β^B, γ^B) in $\mathcal{O}(L^5 \log L + N_{\mathbf{R}^B} L^3)$ steps.

4. Compute

$$C = \sum_{\ell=0}^{L-1} \sum_{m=-\ell}^{\ell} \sum_{n=-\ell}^{\ell} \sum_{k=\ell+1}^{L} \hat{a}_{k\ell m} \hat{e}_{k\ell n}(-\Delta z, \beta^B, \gamma^B) D_\ell^{n,m}(\mathbf{R}^A)$$

using Recipe 1 in $\mathcal{O}(N_{\mathbf{R}^B}(L^4 + L^3 \log L + N_{\mathbf{R}^A}))$ steps.

The overall cost is $\mathcal{O}(L^6 + L^5 \log L + N_{\mathbf{R}^B}(L^3 + L^4) + N_{\mathbf{R}^B} N_{\mathbf{R}^A}) N_T$, or $\mathcal{O}(L^6 + L^4 N_{\mathbf{R}^B} + N_{\mathbf{R}^B} N_{\mathbf{R}^A}) N_t$. □

Proof of Theorem 6.4.

The translation coefficients $T^{|m|}_{k'l',kl}(z) \cdot \exp(z^2/4\lambda)$ are polynomials of degree

$$\max(n + 2M) = \max(n + 2(j + \frac{l + l' - k}{2})) = \max(2j + l + l') = 2k - l + 2k' - l' - 4.$$

Let $d = 2k - l + 2k' - l' - 4$, $n = \min(p, l + l') - s$ and $i = \frac{p-n}{2}$ Then Equation 3.12 can be arranged to obtain

$$T^{|m|}_{k'l',kl}(z) \exp(z^2/4\lambda) = \sum_{p=0}^{2k-l+2k'-l'-4} \alpha_p \cdot z^p$$

where

$$\alpha_p = \sum_{s=0}^{\min(p,l+l')-|l-l'|} A_n^{ll'|m|} \sum_{j=\max(i-\frac{l+l'-n}{2},0)}^{k-l+k'-l'-2} C_j^{k,l,k'l'} M! \frac{(1/2)_{M+n+1}}{(M-i)!(1/2)_{n+i+1}} \cdot \frac{1}{(-4\lambda)^i \cdot i!},$$

and s is even iff d is even.

α_p can be computed for all p in $O(L^3)$ steps. For fixed k, l, k', l', m, the T-Matrix polynomial can be computed in $\mathcal{O}(LN_t)$. The complexity for fixed k, l, k', l', m is hence $\mathcal{O}(L^3 + LN_t)$, resulting in an overall complexity of $\mathcal{O}(L^8 + L^6 N_t)$.

A polynomial can be evaluated at a set of equispaced arguments with $\mathcal{O}(L)$ multiplications. Applying Nuttall's update rule for polynomials [23] reduces these multiplications to additions without altering the number of operations required. This affords a small speedup.

Speedup By Dynamic Programming. If $A_n^{ll'|m|}$ is precomputed for all m, the other terms in Equation 3.12 have to be calculated only once for fixed k, l, k', l'. In the first step, we compute

$$b_n := \sum_{j=0}^{k-l+k'-l'-2} C_j^{kl,k'l'} M! \exp(-z^2/4\lambda)\ (z^2/4\lambda)^{n/2} L_M^{(n+1/2)}(z^2/4\lambda)$$

for all m and fixed n, l, n', l'. The summation over j and the computation of $L_M^{n+1/2}$ each takes $\mathcal{O}(L)$ steps, implying a complexity of $\mathcal{O}(L^2)$ for each b_n, and a complexity for all m of $\mathcal{O}(L^3)$.

In the second step we compute the T-Matrix entries

$$T_{k'l',kl}^{|m|} = \sum_{n=|l-l'|}^{l+l'} b_n \cdot A_n^{ll'|m|}.$$

Since the above calculation has to be done for all k, l, k', l' and for N_t translations, the overall complexity for $T_{k'l',kl}^{|m|}$ is now $O(L^7 N_t)$, instead of $O(L^8)$.

Another speed-up can be obtained regarding the $C_j^{kl,k'l'}$. Only these coefficients and the boundary of the innermost sum depend on k and k'. If k and k' are switched, the boundary of the sum does not change, so for switched k and k' only the value $C_j^{kl,k'l'}$ changes. In the first step

$$l(z^2/4\lambda) := L_M^{(n+1/2)}(z^2/4\lambda)$$

is computed for all j, n, l, l'. In the second step

$$t_{kl,k'l'} := \sum_{j=0}^{k-l+k'-l'-2} C_j^{kl,k'l'} M! \exp(-z^2/4\lambda)\ (z^2/4\lambda)^{n/2} l(z^2/4\lambda)$$

and

$$t_{k'l,kl'} := \sum_{j=0}^{k-l+k'-l'-2} C_j^{k'l,kl'} M! \exp(-z^2/4\lambda)\ (z^2/4\lambda)^{n/2} l(z^2/4\lambda)$$

is computed. In the third step

$$T_{k'l',kl}^{|m|} = \sum_{n=|l-l'|}^{l+l'} A_n^{ll'|m|} \cdot t_{kl,k'l'}$$

and

$$T_{kl',k'l}^{|m|} = \sum_{n=|l-l'|}^{l+l'} A_n^{ll'|m|} \cdot t_{k'l,kl'}$$

are computed.

Moreover, the symmetry property [30]

$$T^{|m|}_{k'l',kl} = (-1)^{l-l'} T^{|m|}_{kl,k'l'}$$

implies

$$T^{|m|}_{kl',k'l} = (-1)^{l-l'} T^{|m|}_{k'l,kl'}.$$

The dynamic programming approach above allows us to calculate $T^{|m|}_{kl,k'l'}$, $T^{|m|}_{kl',k'l}$ and $T^{|m|}_{k'l,kl'}$ by calculating $T^{|m|}_{k'l',kl}$.

Combining The Different Speed-Ups. The complexity of the approach of representing the T-coefficients as a polynomial can be reduced by using the speed-up by dynamic programming as explained above. To achieve the reduction in the complexity we consider the calculation of α_p. Instead of computing α_p directly, first

$$b^p_s := \frac{1}{(-4\lambda)^i \cdot i!(1/2)_{n+i+1}} \sum_{j=\max(i-\frac{l+l'-n}{2},0)}^{k-l+k'-l'-2} C_j^{k,l,k'l'} M! \frac{(1/2)_{M+n+1}}{(M-i)!}$$

is precomputed. This computation has the complexity $O(L^3)$, because of the summation and the parameters s and p. Afterwards the α_p

$$\alpha_p = \sum_{s=0}^{\min(p,l+l')-|l-l'|} A_n^{ll'|m|} \cdot b^p_s.$$

are computed. This has the complexity $\mathcal{O}(L^2)$, implying a complexity of $\mathcal{O}(L^3)$ for the precomputation of α_p for all m. The total computation of the α_p for all m is hence $\mathcal{O}(L^3 + L^3) = \mathcal{O}(L^3)$.

The subsequent computation of

$$T^{|m|}_{kl',k'l} \exp(z^2/4\lambda) = \sum_{p=0}^{2k-l+2k'-l'-4} \alpha_p \cdot z^p$$

is for fixed k, l, k', l', m and all m is $O(L^2 N_t)$. Therefore the overall complexity for fixed k, l, k', l' and all m is $\mathcal{O}(L^3 + L^2 N_t)$. Thus, for all k, l, k', l' the complexity is $\mathcal{O}(L^4)O(L^3 + L^2 N_t) = \mathcal{O}(L^7 + L^6 N_t)$ $\square$.

References.

[1] A. Abyzov, R. Bjornson, M. Felipe, and M. Gerstein. RigidFinder: A fast and sensitive method to detect rigid blocks in large macromolecular complexes. *PROTEINS: Structure, Function, and Bioinformatics*, 78:309–324, 2010.

[2] C. Bajaj, R. Chowdhury, and V. Siddahanavalli. F2Dock: Fast fourier protein-protein docking. *IEEE/ACM Trans. Comput. Biol. Bioinf*, 8(1):45–58, 2011.

[3] C. Bajaj, R. A. Chowdhury, and V. Siddavanahalli. F3dock: A fast, flexible and fourier-based approach to protein-protein docking. Technical report, The University of Texas, 2007.

[4] R. Bettadapura, A. Vollrath, and C. Bajaj. Pfflexfit: Hierarchical flexible fitting in 3d em. Technical Report 12-29, University of Texas at Austin, July 2012.

[5] L. Biedenharn and J. Louck. *The Racah-Wigner algebra in quantum theory.* Addison-Wesley, 1981.

[6] G. Chirikjian and I. Ebert-Uphoff. Numerical convolution on the euclidean group with applications to workspace generation. *IEEE Trans. on Robotics and Automation*, 14(1):123–136, 1998.

[7] G. S. Chirikjian and A. Kyatkin. *Engineering applications of noncommutative harmonic analysis with emphasis on rotation and motion groups.* CRC Press, Boca Raton, 2001.

[8] M. Danos and L. C. Maximon. Multipole matrix elements of the translational operator. *Journal of Mathematical Physics*, 6(5):766–778, 1965.

[9] S. Flores, L. Lu, J. Yang, N. Carriero, and M. Gerstein. Hinge atlas: relating protein sequence to sites of structural flexibility. *BMC Bioinformatics*, 8:167–186, 2007.

[10] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE. Invited paper, Special Issue on Program Generation, Optimization, and Platform Adaptation*, 93(2):216–231, February 2005.

[11] J. I. Garçon, J. A. Kovacs, and R. Abagyan. Adp_em: Fast exhaustive multiresolution docking with high-throughput coverage. *Bioinformatics*, 23(4):427–433, 2007.

[12] L. Holm and C. Sander. Parser for protein folding units. *Proteins*, 19(3):256–268, July 1994.

[13] L. Kale, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. Namd2: Greater scalability for parallel molecular dynamics. *J. Comput. Phys.*, 151(1):283–312, 1999.

[14] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and I. A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences of the United States of America*, 89(6):2195–2199, 1992.

[15] J. Keiner, S. Kunis, and D. Potts. Using nfft 3 - a software library for various nonequispaced fast fourier transforms. *ACM Transactions on Mathematical Software (TOMS)*, 36(4):19:1–19:30, 2009.

[16] J. A. Kovacs, P. Chacón, Y. Cong, E. Metwally, and W. Wriggers. Fast rotational matching of rigid bodies by fast fourier transform acceleration of five degrees of freedom. *Acta Crystallographica Section D*, D59:1371–1376, 2003.

[17] J. A. Kovacs and W. Wriggers. Fast rotational matching. *Acta Crystallographica Section D*, 58(8):1282–1286, Aug 2002.

[18] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 238–249, 1988.

[19] J. Leech, J. Prins, and J. Hermans. Smd: visual steering of molecular dynamics for protein design. *IEEE Computational Science and Engineering*, 3(4):38–45, 1996.

[20] V. N. Maiorov and R. A. Abagyan. A new method for modeling large-scale rearrangements of protein domains. *Proteins*, 27:410–424, 1997.

[21] J. G. Mandell, V. A. Roberts, M. E. Pique, V. Kotlovyi, J. C. Mitchell, E. Nelson, I. Tsigelny, and L. F. T. Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering Design and Selection*, 14(2):105–113, 2000.

[22] J. C. Mitchell. Discrete uniform sampling of rotation groups using orthogonal images. *SIAM Journal of Scientific Computing*, 30(1):525–547, 2007.

[23] A. H. Nuttall. Efficient evaluation of polynomials and exponentials of polynomials for equispaced arguments. *IEEE Transactions On Acoustics, Speech And Signal Processing*, 35(10):1486–1487, 1987.

[24] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. Ucsf chimera–a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–12, 2004.

[25] B. G. Pierce, Y. Hourai, and Z. Weng. Accelerating protein docking in zdock using an advanced 3d convolution library. *PLoS One*, 6(9), 2011.

[26] G. Poornam, A. Matsumoto, H. Ishida, and S. Hayward. A method for the analysis of domain movements in large biomolecular complexes. *PROTEINS: Structure, Function, and Bioinformatics*, 76:201–212, 2009.

[27] D. Potts, J. Prestin, and A. Vollrath. A fast algorithm for nonequispaced fourier transforms on the rotation group. *Numerical Algorithms*, 52(3):355–384, 2009.

[28] D. Potts, G. Steidl, and M. Tasche. Fast algorithms for discrete polynomial transforms. *Math. Comput.*, 67:1577 – 1590, 1998.

[29] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, chapter 12, pages 247 – 270. Birkhäuser, Boston, 2001.

[30] D. Ritchie. High-order analytic translation matrix elements for real-space six-dimensional polar fourier correlations. *Journal of Applied Crystallography*, 38:808–818, 2005.

[31] D. Ritchie and G. Kemp. Protein docking using spherical polar fourier correlations. *Proteins, Structure, Function and Genetics*, 39, 2000.

[32] D. W. Ritchie, D. Kozakov, and S. Vajda. Accelerating and focusing protein-protein docking correlations using multi-dimensional rotational fft generating functions. *Bioinformatics*, 24(17):1865–1873, 2008.

[33] M. Rossmann. The molecular replacement method. *Acta Crystallogr. Sect. A*, 46:73–82, 1990.

[34] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson. Geometry-based flexible and symmetric protein docking. *Proteins: Structure, Function, and Bioinformatics*, 60(2):224–231, 2005.

[35] M. Shatsky, R. Nussinov, and H. Wolfson. Flexible protein alignment and hinge detection. *PROTEINS: Structure, Function, and Genetics*, 48:242–256, 2002.

[36] R. D. Skeel, I. Tezcan, and D. J. Hardy. Multiple grid methods for classical molecular dynamics. *Journal of Computatioanl Chemistry*, 23(6):673–684, 2002.

[37] SourceForge. Kissfft. `http://sourceforge.net/projects/kissfft/`, 2012.

[38] J. E. Stone, J. Gullingsrud, and K. Schulten. A system for interactive molecular dynamics simulation. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 191–194. ACM Press, 2001.

[39] G. Szegő. *Orthogonal Polynomials*. Amer. Math. Soc., Providence, 4th edition, 1975.

[40] M. Topf, M. L. Baker, B. John, W. Chiu, and A. Sali. Structural characterization of components of protein assemblies by comparative modeling and electron cryo-microscopy. *Journal of Structural Biology*, 149:191–203, 2005.

[41] M. Topf, K. Lasker, B. Webb, H. Wolfson, W. Chiu, and A. Sali. Protein structure fitting and refinement guided by cryoem density. *Structure*, 16(2):295–307, 2008.

[42] L. G. Trabuco, E. Villa, K. Mitra, J. Frank, and K. Schulten. Flexible fitting

of atomic structures into electron microscopy maps using molecular dynamics. *Structure*, 2008.

[43] N. Woetzel, S. Lindert, P. Stewart, and J. Meiler. Bcl::em-fit: Rigid body fitting of atomic structures into density maps using geometric hashing and real space refinement. *Journal of Structural Biology*, 3(264-76), 2011.

[44] W. Wriggers. Using situs for the integration of multi-resolution structures. *Biophysical Reviews*, 2(1):21–27, 2010.

[45] W. Wriggers and P. Chacon. Multi-resolution contour-based fitting of macromolecular structures. *Journal of Molecular Biology*, 317:375–384, 2002.

[46] W. Wriggers, R. A. Milligan, K. Schulten, and J. A. McCammon. Self-organizing neural networks bridge the biomolecular resolution gap. *Journal of Molecular Biology*, 287(1247-1254), 1998.

[47] Q. Zhang, R. Bettadapura, and C. Bajaj. Macromolecular structure modeling from 3dem using volrover 2.0. *Biopolymers*, 97(9):709–731, 2012.

[48] W. Zheng. Accurate flexible fitting of high-resolution protein structures into cryo-electron microscopy maps using coarse-grained pseudo-energy minimization. *Biophysical Journal*, 100:478–488, 2011.