Synchronizing Probability Measures on Rotations via Optimal Transport Supplementary Document

Tolga Birdal [1] Michael Arbel [2] Umut Şimşekli [3, 4] Leonidas Guibas [1]
[1] Department of Computer Science, Stanford University, USA
[2] Gatsby Computational Neuroscience Unit, University College London, UK
[3] LTCI, Télécom Paris, Institut Polytechnique de Paris, Paris, France
[4] Department of Statistics, University of Oxford, Oxford, UK

Abstract

This document supplements our main paper entitled Synchronizing Probability Measures on Rotations via Optimal Transport. In specifics, we provide a background section on the technical part, the explicit form of the gradients required by the algorithm, a more detailed explanation on our assumptions and the composition functions. We also include the pseudocode of our method and two additional experiments: (1) on a real SfM dataset, (2) on the mug sequence shown in the main paper.

1. Connection to Maximum Likelihood Estimation (MLE) and Markov Random Fields (MRF)

Rotation synchronization has been studied in the literature under the name *multiple rotation averaging*. The standard single-particle based methods such as SE-Sync [13] assume a unimodal Gaussian/Langevin distribution. There are two caveats with that. First, the classical approaches cannot yield explicit uncertainty estimates, and second a unimodal distribution cannot capture ambiguities that can be multimodal. DISCO has tackled this problem via MRFs and loopy belief propagation [7]. In fact our formulation is similar when the nodes are assumed to have uniform prior. Yet, like K-best synchronization [15], DISCO requires a single pairwise potential, as opposed to the multimodal distributions we have. To the best of our knowledge, such MRF methods have not been extended to work in our setting. Note that differently to all those our approach falls in the non-parametric inference.

2. Optimal Transport on Riemannian Manifolds

Here we denote by $\mathbf{X}$ a Riemannian manifold, which can be for instance the set of unit quaternions $\mathbb{H}$.

2.1. Optimal Transport

For two given probability distributions ν and μ in $\mathcal{P}_2(\mathbf{X})$, we denote by $\Pi(\nu,\mu)$ the set of couplings between ν and μ, i.e.: $\Pi(\nu,\mu)$ contains all joint distributions π on $\mathbf{X}\times\mathbf{X}$ such that if $(X,Y)\sim\pi$ then $X\sim\nu$ and $Y\sim\mu$. The 2-Wasserstein distance on $\mathcal{P}_2(\mathbf{X})$ is defined by means of an optimal coupling between ν and μ:

$$W_2^2(\nu,\mu) := \inf_{\pi\in\Pi(\nu,\mu)} \int \|x-y\|^2 \,\mathrm{d}\pi(x,y) \qquad \forall \nu,\mu\in\mathcal{P}_2(\mathbf{X}) \tag{1}$$

It is a well established fact that such optimal coupling π^* exists [16, 14] . Moreover, W_2 enjoys a dynamical formulation which gives it an interpretation as the length of the shortest path connecting ν and μ in probability space. It is summarized by the celebrated Benamou-Brenier formula ([1]):

$$W_2(\nu,\mu) = \inf_{(\rho_t,V_t)_{t\in[0,1]}} \int_0^1 \int \|V_t(x)\|^2 d\rho_t(x), \tag{2}$$

where the infimum is taken over all couples ρ and v satisfying a continuity equation with boundary conditions:

$$\partial_t \rho_t + \mathrm{div}(\rho_t V_t) = 0, \qquad \rho_0 = \nu, \quad \rho_1 = \mu. \tag{3}$$

The above equation expresses two facts, the first one is that $-\mathrm{div}(\rho_t V_t)$ reflects the infinitesimal changes in ρ_t as dictated by the vector field (also referred to as velocity field) V_t, the second one is that the total mass of ρ_t does not vary in time as a consequence of the divergence theorem. Equation Eq (3) is well defined in the distribution sense even when ρ_t does not have a density and V_t can be interpreted as a tangent vector to the curve $(\rho_t)_{t\in[0,1]}$.

In Sec. 2.4 we will see that the continuity equation in Eq (10) without terminal condition $\rho_1 = \mu$ and for a well chosen vector field V_t leads to a gradient flow in probability space.

2.2. First variation of a functional

Here we introduce the notion of first variation of a functional $\mathcal{F}$ which will be crucial to define the Wasserstein gradient flow in Sec. 2.3. We then provide explicit expressions of this first variation in the case of the MMD and sinkhorn divergence.

Consider a real valued functional $\mathcal{F}$ defined over $\mathcal{P}_2(\mathbf{X})$. We call $\frac{\partial \mathcal{F}}{\partial \nu}$ if it exists, the unique (up to additive constants) function such that $\frac{d}{d\epsilon}\mathcal{F}(\nu + \epsilon(\nu' - \nu))|_{\epsilon=0} = \int \frac{\partial \mathcal{F}}{\partial \nu}(\nu)(\mathrm{d}\nu' - \mathrm{d}\nu)$ for any $\nu' \in \mathcal{P}_2(\mathbf{X})$. For a fixed ν, the function $\frac{\partial \mathcal{F}}{\partial \nu}(\nu)$ is a real valued function defined on $\mathbf{X}$ and is called the first variation of $\mathcal{F}$ evaluated at ν.

In the case of the squared MMD, a simple expression is obtained by direct calculation:

$$\frac{\partial \mathrm{MMD}^2(\nu,\mu)}{\partial \nu}(\nu)(x) = 2\left(\int k(x,y)d\nu(y) - \int k(x,y)d\mu(y)\right). \tag{4}$$

This can be easily estimated using samples from both μ and ν.

The first variation of the Sinkhorn is more involved. We first recall the expression of the Sinkhorn distance $d_{c,\alpha}(\mu,\nu)$ in terms of the optimal potential functions:

$$d_{c,\alpha}(\mu,\nu) = \int f(x)d\mu(x) + \int g(x)d\nu(x) \tag{5}$$

where f and g are unique up to an additive constant [9, Proposition 1]. In practice, given samples $(X_i)_{1\leq i\leq N}$ and $(Y_i)_{1\leq i\leq N}$ from ν and μ, f and g can be estimated on those values using the iterative sinkhorn algorithm, this provides vectors f_i and g_i such that $f_i \sim f(X_i)$ and $g_i \sim g(Y_i)$.

The first variation of the Sinkhorn distance is simply given by differentiating wrt ν:

$$\frac{\partial d_{c,\epsilon}(\mu,\nu)}{\partial \nu}(\nu)(x) = g(x) \tag{6}$$

However, g needs to be evaluated at arbitrary points x, while the Sinkhorn algorithm only provides the values g_i and f_i at the sample points X_i and Y_i. This is not an issue as noted in [10, 9]. Indeed, f and g are related by the equation:

$$g(x) = -\epsilon \log\left(\int \exp\left(\frac{f(y) - c(x,y)}{\epsilon}\right)d\nu(y)\right) \tag{7}$$

Hence, g can be estimated by replacing the expectation by the empirical one and using the estimated values f_i at the sample points Y_i :

$$\hat{g}(x) = -\epsilon \log\left(\frac{1}{N}\exp\left(\frac{f_i - c(x,Y_i)}{\epsilon}\right)\right). \tag{8}$$

Finally, the variation of the Sinkhorn divergence, is obtained by summing those of each of it's components:

$$\frac{\partial \mathcal{S}_{c,\epsilon}(\mu,\nu)}{\partial \nu}(\nu)(x) = 2 * \frac{\partial d_{c,\epsilon}(\mu,\nu)}{\partial \nu}(\nu)(x) - \frac{\partial d_{c,\epsilon}(\nu,\nu)}{\partial \nu}(\nu)(x). \tag{9}$$

2.3. Wasserstein gradient flow

The formal gradient flow equation associated to a functional $\mathcal{F}$ can be written (see [4], Lemma 8 to 10):

$$\frac{\partial \nu_t}{\partial t} = \mathrm{div}\left(\nu_t \nabla \frac{\partial \mathcal{F}}{\partial \nu_t}\right) \tag{10}$$

where div is the divergence operator and $\nabla \frac{\partial \mathcal{F}}{\partial \nu}(x)$ is the Riemannian gradient of $\frac{\partial}{\partial \nu}\mathcal{F}(x)$ which is an element of the tangent space of $\mathbf{X}$ at point x. It can be shown that the probability distributions ν_t decrease $\mathcal{F}$ in time. More precisely the following energy dissipation equation holds under mild regularity conditions on $\mathcal{F}$:

$$\mathcal{F}(\nu_t) = -\int \|\nabla \frac{\partial \mathcal{F}}{\partial \nu}(\nu_t)(x)\|^2 d\nu_t(x). \tag{11}$$

$\mathcal{F}(\nu_t)$ is a decreasing function in time, hence the interpretation of ν_t as a gradient flow of $\mathcal{F}$.

2.4. Riemannian Particle Descent

The equation in 10 admits an equivalent expression in terms of particles which will be useful in practice:

$$\frac{X_t}{dt} = -\nabla \frac{\partial \mathcal{F}}{\partial \nu}(\nu_t)(X_t) \tag{12}$$

A discretization in time and space can be performed in the following way: Given N initial particles $(X_0^i)_{1\leq i\leq N}$, and a step-size γ, the following update rule can be used:

$$X_{t+1}^i = \exp_{X_t^i}(-\gamma \nabla \frac{\partial \mathcal{F}}{\partial \nu}(\hat{\nu}_t)(X_t^i)) \tag{13}$$

where $\hat{\nu}_t$ is the particle measure at time t: $\hat{\nu}_t = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^i}$ and exp is the exponential map associated to the manifold $\mathbf{X}$. We refer to Sec. 3.1 for a closed form expression of the exponential map in the case of unit-quaternions. We provide a pseudocode for the proposed RPGD algorithm in Algorithm 1. We also release our implementation under: https://synchinvision.github.io/probsync.

Algorithm 1: Riemannian Particle Gradient Descent for Measure Synchronization

input : Relative measures $\{\boldsymbol{\mu}_{ij}\}_{i,j=1}^n$
output: Absolute measures $\{\boldsymbol{\mu}_i\}_{i=1}^n$
// Initialize the particles
$\mathbf{q}_i^{(k)} \sim \boldsymbol{\mu}_i, \qquad i = 1,\dots,n, \quad k = 1,\dots,K_n$
// Iterations
for $t = 0,\dots T-1$ **do**
  // For all cameras
  for $i = 1,\dots,n$ **do**
    // Update the positions of the particles
    $\mathbf{q}_i^{(k)} \leftarrow Exp_{\mathbf{q}_i^{(k)}}\Big(-\eta_q(w_i^{(k)})\nabla_{\mathbf{q}_i^{(k)}}\mathcal{L}(\boldsymbol{\mu})\Big) \qquad k = 1,\dots,K_n$
    // Update the weights of the particles -- Unconstrained case
    $\beta_i^{(k)} \leftarrow \beta_i^{(k)} - \eta_\beta \nabla_{\beta_i^{(k)}}\big(\mathcal{L}(\boldsymbol{\mu}) + \mathcal{R}(\boldsymbol{\mu})\big) \qquad k = 1,\dots,K_n$
    // Update the weights of the particles -- Constrained case
    $\beta_i^{(k)} \leftarrow Exp_{\beta_i^{(k)}}\Big(-\eta_\beta \nabla_{\beta_i^{(k)}}\big(\mathcal{L}(\boldsymbol{\mu})\big)\Big)$

3. Analytic Form of the Gradients

In this section, we provide the analytical forms of the gradients required by our algorithm. We first recall the expression of the normalized logarithm of a unit quaternion $\mathbf{x} := (a, \mathbf{v})$ which is given by:

$$\frac{\log(\mathbf{x})}{\|\mathbf{x}\|} = (0, \frac{\mathbf{v}}{\|\mathbf{v}\|}) \tag{14}$$

we also right $\log_{\mathbf{x}}(\mathbf{y}) := \log(\mathbf{x}^{-1}\mathbf{y})$. A subgradient of the Riemannian distance $d(\cdot)$ is given as follows:

$$\nabla_{\mathbf{x}} d(\mathbf{x}\in\mathbb{H}, \mathbf{y}\in\mathbb{H}) = \begin{cases} -\operatorname{sign}(\langle \mathbf{x}, \mathbf{y}\rangle)\frac{\log_{\mathbf{x}}(\mathbf{y})}{\|\log_{\mathbf{x}}(\mathbf{y})\|} \equiv \left(0, -s\frac{\mathbf{v}}{\|\mathbf{v}\|}\right) & \mathbf{x}\neq\mathbf{y} \\ 0 & \mathbf{x}=\mathbf{y} \end{cases} \tag{15}$$

where $\mathbf{v}$ denotes the imaginary part of $\mathbf{x}^{-1}\mathbf{y}$ and $s := \text{sign}(\langle \mathbf{x}, \mathbf{y} \rangle)$ is the sign of dot product between $\mathbf{x}$ and $\mathbf{y}$ with the convention that $s = 1$ if the dot product is 0.

By using this formulation and the chain rule of differentiation, we obtain the gradient required by RPGD which is given by Proposition 1. We finally combine this gradient with the gradient of the Sinkhorn divergence or the gradient of MMD by using autodiff.

Proposition 1. *The gradient of $d(\mathbf{q}_i\mathbf{q}_j^{-1}, \mathbf{q}_{ij})$ w.r.t. $\mathbf{q}_i$ and $\mathbf{q}_j$ is given by:*

$$\nabla_{\mathbf{q}_i} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij})) = \mathbf{q}_j^{-1} \nabla_{\hat{\mathbf{q}}_{ij}} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) \mathbf{q}_j \tag{16}$$

$$\nabla_{\mathbf{q}_j} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = -\nabla_{\mathbf{q}_i} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) \tag{17}$$

where $\hat{\mathbf{q}}_{ij} = \mathbf{q}_i\mathbf{q}_j^{-1}$

Proof. First recall that d is bi-invariant, hence:

$$d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = d(\mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_j)$$

Excluding the case when $\mathbf{q}_i = \mathbf{q}_{ij}\mathbf{q}_j$ (for which the expression is trivial), we have that:

$$\nabla_{\mathbf{q}_i} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = \nabla_{\mathbf{q}_i} d(\mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_j) = -\text{sign}(\langle \mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_j \rangle) \frac{\log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j)}{\| \log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j) \|} \tag{18}$$

It is easy to see that $\langle \mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_i \rangle = \langle \hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij} \rangle$ since composition of the two rotations $\mathbf{q}_i$ and $\mathbf{q}_{ij}$ by $\mathbf{q}_j$ preserves the angles. On the other hand, one can observe that $\mathbf{q}_i^{-1}\mathbf{q}_{ij}q_j = \mathbf{q}_j^{-1}\hat{\mathbf{q}}_{ij}^{-1}\mathbf{q}_{ij}\mathbf{q}_j$ and apply Lemma 1 to get:

$$\frac{\log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j)}{\| \log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j) \|} = \frac{\log(\mathbf{q}_j^{-1}\hat{\mathbf{q}}_{ij}^{-1}\mathbf{q}_{ij}\mathbf{q}_j)}{\| \log(\mathbf{q}_j^{-1}\hat{\mathbf{q}}_{ij}^{-1}\mathbf{q}_{ij}\mathbf{q}_j) \|} = \mathbf{q}_j^{-1} \frac{\log_{\hat{\mathbf{q}}_{ij}}(\mathbf{q}_{ij})}{\| \log_{\hat{\mathbf{q}}_{ij}}(\mathbf{q}_{ij}) \|} \mathbf{q}_j, \tag{19}$$

This shows the first identity. The second identity is obtained similarly. By bi-invariance of d, we have that $d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = d(\mathbf{q}_j, \mathbf{q}_{ij}^{-1}\mathbf{q}_i)$, hence:

$$\nabla_{\mathbf{q}_j} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = \nabla_{\mathbf{q}_j} d(\mathbf{q}_j, \mathbf{q}_{ij}^{-1}\mathbf{q}_i) = -\text{sign}(\langle \mathbf{q}_j, \mathbf{q}_{ij}^{-1}\mathbf{q}_i \rangle) \frac{\log_{\mathbf{q}_j}(\mathbf{q}_{ij}^{-1}\mathbf{q}_i)}{\| \log_{\mathbf{q}_j}(\mathbf{q}_{ij}^{-1}\mathbf{q}_i) \|}. \tag{20}$$

Moreover, we have that $\mathbf{q}_j^{-1}\mathbf{q}_{ij}^{-1}\mathbf{q}_i = (\mathbf{q}_i^{-1}\mathbf{q}_{ij}\mathbf{q}_j)^{-1}$, thus using that $\log(\mathbf{x}^{-1}) = -\log(\mathbf{x})$ and that $\langle \mathbf{q}_j, \mathbf{q}_{ij}^{-1}\mathbf{q}_i \rangle = \langle \mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_j \rangle$ it follows :

$$\nabla_{\mathbf{q}_j} d(\hat{\mathbf{q}}_{ij}, \mathbf{q}_{ij}) = \text{sign}(\langle \mathbf{q}_i, \mathbf{q}_{ij}\mathbf{q}_j \rangle) \frac{\log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j)}{\| \log_{\mathbf{q}_i}(\mathbf{q}_{ij}\mathbf{q}_j) \|}. \tag{21}$$

which concludes the proof. □

Lemma 1. *Let x and q be unit quaternions, then the following holds:*

$$\frac{\log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q})}{\| \log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q}) \|} = \mathbf{q}^{-1} \frac{\log(\mathbf{x})}{\| \log(\mathbf{x}) \|} \mathbf{q} \tag{22}$$

$$\tag{23}$$

Proof. Let's first prove the first equality, we write $\mathbf{x} = (b, w)$ and $\mathbf{q} = (c, v)$ where b and c are the real parts of $\mathbf{x}$ and $\mathbf{q}$ while w and v are their complex part. by definition of the quaternion product, we have that:

$$\mathbf{q}^{-1}\mathbf{x}\mathbf{q} = (b, (c^2 - \|v\|^2)w + 2\langle v, w \rangle v + 2cw \wedge v) \tag{24}$$

Let's call $Z = (c^2 - \|v\|^2)w + 2\langle v, w \rangle v + 2cw \wedge v$ to simplify notations. Hence, we have by definition of the logarithm:

$$\frac{\log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q})}{\| \log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q}) \|} = (0, \frac{Z}{\|Z\|}) \tag{25}$$

On the other hand, we also have that $\frac{\log(\mathbf{x})}{\|\log(\mathbf{x})\|} = (0, \frac{w}{\|w\|})$, hence:

$$\mathbf{q}^{-1}\frac{\log(\mathbf{x})}{\|\log(\mathbf{x})\|}\mathbf{q} = (0, \frac{1}{\|w\|}\left(c^2 - \|v\|^2\right)w + 2\langle v, w\rangle v + 2cw \wedge v) := (0, \frac{Z}{\|w\|})$$

We have shown that $\mathbf{q}^{-1}\frac{\log(\mathbf{x})}{\|\log(\mathbf{x})\|}\mathbf{q}$ and $\frac{\log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q})}{\|\log(\mathbf{q}^{-1}\mathbf{x}\mathbf{q})\|}$ have the same direction, since both are unit vectors, they must be equal. □

3.1. Exponential map in quaternion space

We provide a closed form expression for the exponential map used in the update rule: let q be an element in the unit quaternion manifold, i.e: $\|q\| = 1$, and v an element of it's tangent space which is necessarily of the form $v = (0, w)$ where w is a vector in $\mathbb{R}^3$. Indeed, a vanishing first component insures that the v doesn't contain components that are orthogonal to the unit quaternion manifold.

$$exp_q(v) = q\exp(v) := q(cos(\|w\|), sin(\|w\|)\frac{w}{\|w\|}) \tag{26}$$

4. The Composition Function

- High entropy: In this case g_{ij} is given by:

$$g_{ij}(\boldsymbol{\mu}) = \sum_{k_i=1}^{K_i}\sum_{k_j=1}^{K_j} w_i^{(k_i)} w_j^{(k_j)} \delta_{q_i^{(k_i)}\overline{q_j^{(k_j)}}} \tag{27}$$

- Low entropy In this case, $K_i = K_j = K$ and the weights satisfy the additional constraint: $w_i^{(k)} = w_j^{(k)} = w^{(k)}$ for some non-negative numbers $(w^{(k)})_{1\leq k\leq K}$ that sum to 1. Moreover, we have that:

$$g_{ij}(\boldsymbol{\mu}) = \sum_{k=1}^{K} w^{(k)} \delta_{q_i^{(k)}\overline{q_j^{(k)}}} \tag{28}$$

5. More Details about the Theoretical Result

We start by detailing the assumption **H**3. In particular, we will give the precise definition of a *non-degenerate* minimum. We denote by $\mathbf{q}$ a vector of n quaternions $(\mathbf{q}_1, ..., \mathbf{q}_n)$ and by $(\mathbf{q}^*)^{(k)}$ the k-th particle from the optimal distribution $\boldsymbol{\mu}^*$. We will introduce the same objects as in [6, section 3.1]. Note that in all our setting we fix the ratio of the learning rates $\alpha := \frac{\eta_\beta}{\eta_q}$ to 0.1. Let us first define $\mathcal{J}(\boldsymbol{\mu}) = \mathcal{L}(\boldsymbol{\mu}) + \mathcal{R}(\boldsymbol{\mu})$ and denote by $\mathcal{J}'_{\boldsymbol{\mu}^\star}$ the differential of $\mathcal{J}$ at $\boldsymbol{\mu}^\star$ [6].The *local kernel matrix* H, is defined as a matrix in $\mathbb{R}^{(K*\times(1+4n))^2}$ as follows:

$$H_{(k,l:l+4),(k',l':l'+4)} = \begin{cases} \nabla^2_{\mathbf{q}_l,\mathbf{q}_{l'}}\mathcal{J}'_{\boldsymbol{\mu}^\star}\left((\mathbf{q}^\star)^{(k)}\right) & \text{if } k = k' \text{ and } l, l' \geq 1 \\ 0 & \text{if } l = 0 \text{ or } l' = 0 \end{cases}, \tag{29}$$

where $\nabla^2\mathbf{q}_l, \mathbf{q}_{l'}$ denotes the Hessian matrix that is composed of the partial derivatives $\partial_{\mathbf{q}_l}\partial_{\mathbf{q}_{l'}}$. We also introduce the features $\Phi((\mathbf{q}_1, ..., \mathbf{q}_n))$ and Ψ defined by:

$$\Phi((\mathbf{q}))_{ij} = y \mapsto k(g_{ij}(\mathbf{q}), y)$$
$$\Psi_{ij} = y \mapsto \int k(y', y)\,\mathrm{d}\boldsymbol{\mu}_{ij}(y')$$

for all $(i, j) \in \mathcal{E}$. Hence, the loss function $\mathcal{L}(\boldsymbol{\mu})$ can be re-expressed as:

$$\mathcal{L}(\boldsymbol{\mu}) = \mathcal{N}(\int \Phi(\mathbf{q})\,\mathrm{d}\boldsymbol{\mu}(\mathbf{q})) := \sum_{(i,j)\in\mathcal{E}} \| \int \phi_{ij}(\mathbf{q})\,\mathrm{d}\boldsymbol{\mu}(\mathbf{q}) - \Psi_{ij}\|^2_{\mathcal{H}}. \tag{30}$$

Finally, we define the *global kernel* K given by:

$$K_{(k,l),(k',l')} = \langle \beta_k \bar{\nabla}\Phi((\mathbf{q}^\star)^{(k)}), \beta_{k'} \bar{\nabla}\Phi((\mathbf{q}^\star)^{(k')}) \rangle_{d^2 R_{f^\star}} \tag{31}$$

where β_k are such that the optimal weights $w_k^\star$ satisfy: $w_k^\star = \beta_k^2$, the extended gradient $\bar{\nabla}$ is defined to be $\bar{\nabla}\Phi := (2\alpha\Phi, \nabla\Phi)$ and the inner product is taken w.r.t. hessian of $\mathcal{N}$ at $f^\star = \int \Phi(\mathbf{q})\, \mathrm{d}\boldsymbol{\mu}^\star(\mathbf{q})$ as in [6]. Note that in general K is positive semi-definite, however, as we will see now, **H**3 requires it to be definite. Now, we precise the definition of **H**3 as follows:

H3. *The following conditions hold.*

- *The matrix K is positive definite.*
- *The smallest singular value of H is strictly larger than* 0.
- *The only points where $\mathcal{J}'$ vanishes are the optimal particles $(\mathbf{q}^\star)^{(j)}$.*

Accordingly, we precise the statement of the theorem given in the main paper.

Theorem 1. *Consider the LE setting (28) and Case 2 defined in the main paper (i.e. unconstrained case in Alg. 1). Assume that **H**1-4 hold. Then, for any $0 < \varepsilon \leq 1/2$, there exists $C > 0$ and $\rho \in (0, 1)$, such that the following inequality holds:*

$$\mathcal{J}(\boldsymbol{\mu}^{(\kappa)}) - \mathcal{J}(\boldsymbol{\mu}^\star) \leq \left(\mathcal{J}(\boldsymbol{\mu}^{(0)}) - \mathcal{J}(\boldsymbol{\mu}^\star)\right)(1-\rho)^{\kappa-\kappa_0} \tag{32}$$

where $\kappa = 0, 1, \ldots$ denotes the iterations is a constant, and $\kappa_0 = C/\left(\mathcal{J}(\boldsymbol{\mu}^{(0)}) - \mathcal{J}(\boldsymbol{\mu}^\star)\right)^{2+\epsilon}$.

6. Additional Evaluations

Evaluations on the real 1D-SFM dataset [17] We now evaluate two versions of our algorithm on the common benchmark introduced in 1D-SFM [17]. In particular, we compare our results with Chatterjee and Govindu [5] as well as the Weiszfeld rotation averaging [11]. Our results summarized in Tab. 1 demonstrate that the quality of our single particle version (*Ours - K=1*) matches well with the state of the art. We also use multiple particles ($K = 10$) to model the pose distributions even though we have single observed particle corresponding to the relative rotation. This is in essence similar to K-best synchronization [15] and such approach can explain the uncertainty of the estimates as empirical distributions. The results are shown in the column *Ours - K=10*. We pick the best rotation as the particle that has the maximum weight. To achieve this we use the version where we also optimize for the weights. It is seen that such a K-best scheme can have more chances to find the correct mode and even if we are not using explicit M-estimators yields reduced errors that are almost on par with the most-robust methods like Chatterjee & Govindu [5]. In this evaluation we minimize the p-norm with $p = 1.1$. In fact this finding also aligns well with our theory, where optimizing for weights allows us to find a better minimum: A large number of particles initialized randomly ensures the coverage of all basins of attraction of the loss, while optimizing the weights allows to 'kill' particles in bad local minima, in favor of those near global optima. The classical problem where $k = 1$ only allows one particle which can fall in a bad local minimum and can neither escape nor be killed. This is also the reason why our theorem is not applicable to the classical synchronization. It is noteworthy that in this evaluation we omitted the large scenes as our algorithm is computationally more costly than the algorithms specifically designed to solve the single-particle synchronization problem.

Measure Synchronization on the Mug Sequence We have now evaluated our algorithm on the *mug* object shown in our main paper. To do that, we render the depth image of the 3D CAD model of the mug from various viewpoints. For each viewpoint, we back-project the depth image, creating a partial 3D view. When the handle is invisible, the partial view corresponds to a simple cylinder that is hard to match uniquely. We also store the object rotations for each view. The pose of the first image is set to identity. We then impose the graph structure by connecting each view to the 5-nearest. For each edge, we run the voting based point pair feature matching of [8, 3]. The output of this algorithm are K poses ranked by the voting score. We set $K = 12$ so that the pairwise marginals contain 12 particles. Such a procedure yields a set of diverse poses for pairs where multiple alignments are possible, and distributions with single peaks when there exist exact alignments dominating the voting table. We visualize this in Fig. 1. We then run our algorithm on the obtained distributions and record the median and minimum angular errors measured by the geodesic distance. Our algorithm can match the ground truth pose by an error of 4° while the median error of all particles is around 19°. The latter occurs as we are trying to match the entire

Table 1. Median angular errors on 1D-SFM dataset [17].

1DSFM - Scene	Chatterjee & Govindu [5]	Weiszfeld [11]	Ours - K=1	Ours - K=10
Alamo	2.14	3.57	1.66	1.43
Ellis Island	1.15	1.66	0.86	0.86
Madrid Metropolis	3.08	4.37	4.01	3.50
Montreal Notre Dame	0.71	0.92	0.92	0.96
NYC Library	1.40	2.43	2.12	2.12
Piazza del Popolo	2.62	3.35	2.98	1.26
Roman Forum	1.70	2.11	3.61	5.21
Tower of London	2.45	2.73	2.63	2.69
Vienna Cathedral	4.64	5.14	1.89	1.70
Yorkminister	1.62	2.73	1.89	1.89
Average	2.15	2.90	2.26	2.16

distribution rather than a single best. Note that the ability to characterize the entirety of the possibilities is unique to our approach and as shown in this example, is of practical value. For the details of pairwise pose estimation we refer the reader to [8, 3]. Any other pairwise registration algorithm could have been used to obtain the relative rotations provided that multiple potentially uncorrelated solutions can be obtained. In this regard, ICP [2]-like algorithms such as FGR [18] or algorithms that strictly seek a single pose such as [12] are discouraged.

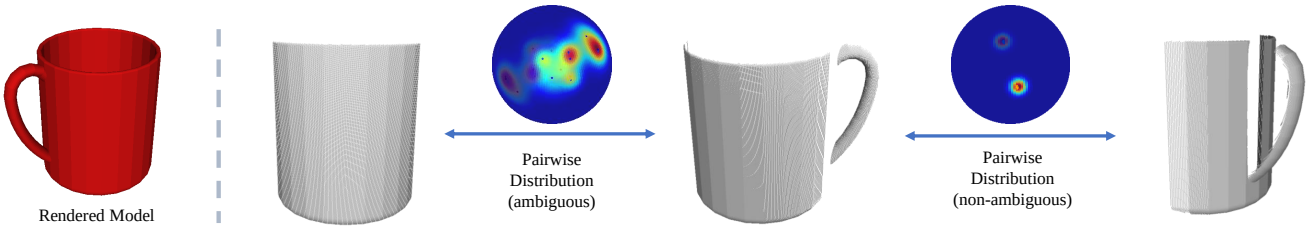

Figure 1. Mug dataset. Each view is back-projected to 3D space creating a partial point cloud. We then estimate multiple possible poses for each pair of points within the vicinity by using a voting based algorithm. This figure shows that such distributions are peaked when objects can be registered uniquely and dispersed when multiple solutions do exist.

References

[1] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. 1

[2] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 7

[3] T. Birdal and S. Ilic. Point pair features based object detection and pose estimation revisited. In *2015 International Conference on 3D Vision*, pages 527–535. IEEE, 2015. 6, 7

[4] J. A. Carrillo, R. J. McCann, and C. Villani. Contractions in the 2-wasserstein length space and thermalization of granular media. *Archive for Rational Mechanics and Analysis*, 179(2):217–263, 2006. 2

[5] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017. 6, 7

[6] L. Chizat. Sparse optimization on measures with over-parameterized gradient descent. *arXiv preprint arXiv:1907.10300*, 2019. 5, 6

[7] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. *IEEE transactions on pattern analysis and machine intelligence*, 35(12), 2012. 1

[8] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. 6, 7

[9] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*. PMLR, 16–18 Apr 2019. 2

[10] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré. Sample complexity of sinkhorn divergences. *arXiv preprint arXiv:1810.02733*, 2018. 2

[11] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3041–3048. IEEE, 2011. 6, 7

[12] N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014. 7

[13] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. Technical Report MIT-CSAIL-TR-2017-002, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, Feb. 2017. 1

[14] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015. 1

[15] Y. Sun, J. Zhuo, A. Mohan, and Q. Huang. K-best transformation synchronization. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2019. 1, 6

[16] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 1

[17] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 6, 7

[18] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016. 7