

# Group Meeting - September 18, 2020

## Paper review & Research progress

Truong Son Hy \*

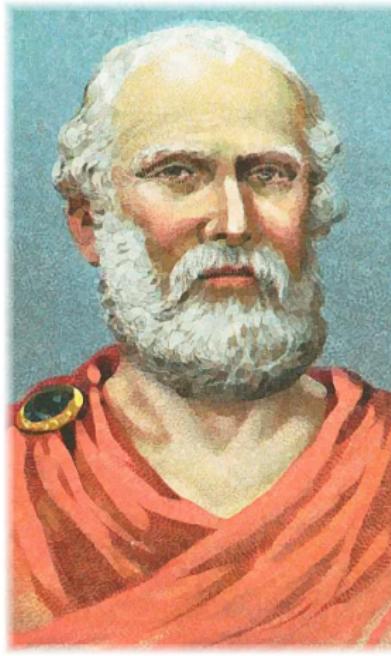
\*Department of Computer Science  
The University of Chicago

Ryerson Physical Lab



# Plato

Never discourage anyone who continually makes progress, no matter how slow.



# Machiavelli

Men judge generally more by the eye than by the hand, for everyone can see and few can feel. Every one sees what you appear to be, few really know what you are.



# Papers – Synchronization

- ① **Probabilistic Permutation Synchronization using the Riemannian Structure of the Birkhoff Polytope** (CVPR 2019)  
<https://arxiv.org/pdf/1904.05814.pdf>
- ② **Synchronizing Probability Measures on Rotations via Optimal Transport** (CVPR 2020)  
<https://arxiv.org/pdf/2004.00663.pdf>
- ③ **Message-passing algorithms for synchronization problems over compact groups**  
<https://arxiv.org/pdf/1610.04583.pdf>

Additional reference for Paper 1:

- ① **Optimization methods on Riemannian manifolds and their application to shape space**  
<https://www.cis.upenn.edu/~cis610/Ring-Wirth-optim-Riemann.pdf>



## Probabilistic Permutation Synchronization using the Riemannian Structure of the Birkhoff Polytope (CVPR 2019)

Tolga Birdal, Umut Şimşekli

<https://arxiv.org/pdf/1904.05814.pdf>



# Proposals

## Proposals

New geometric and probabilistic approach to synchronization of correspondences across multiple sets of objects or images:

- ① **(Geometric)** Birkhoff-Riemannian L-BFGS for optimizing the relaxed version of the combinatorially intractable cycle consistency loss in a principled manner.
- ② **(Geometric)** Birkhoff-Riemannian Langevin Monte Carlo for generating samples on the Birkhoff Polytope and estimating the confidence of the found solutions.
- ③ **(Probabilistic)** Based on Markov Random Field.



# Multi-way graph matching problem

## Map/Permutation Synchronization problem

- The task is to refine the assignments across multiple images/scans (nodes) in a **multi-way graph** and to estimate the assignment confidences.
- Given the correspondences between image pairs as **relative**, **total** permutation matrices, we seek to find **absolute** permutations that re-arrange the detected keypoints to a single canonical, global order.

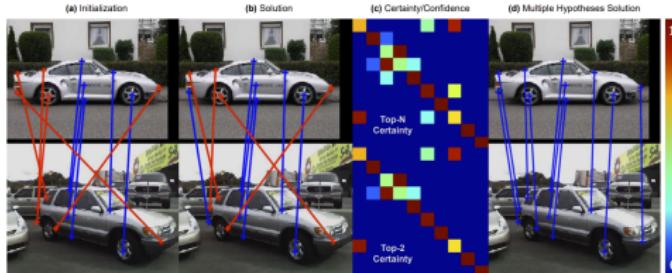


Figure 1. Our algorithm robustly solves the multiway image matching problem (a, b) and provides confidence maps (c) that can be of great help in further improving the estimates (d). The bar on the right is used to assign colors to confidences. For the rest, incorrect matches are marked in red and correct ones in blue.



## Background (1)

### Definition 1 – Permutation Matrix

$$\mathcal{P}_n = \{\mathbf{P} \in \{0, 1\}^{n \times n} : \mathbf{P}\mathbf{1}_n = \mathbf{1}_n, \mathbf{1}_n^T \mathbf{P} = \mathbf{1}_n^T\}$$

Each  $\mathbf{P} \in \mathcal{P}_n$  is a total permutation matrix and  $P_{ij} = 1$  implies that element  $i$  is mapped to element  $j$ .

### Definition 2 – Center of Mass

The center of mass for all the permutations on  $n$  objects is defined as:

$$\mathbf{C}_n = \frac{1}{n!} \sum_{\mathbf{P}_i \in \mathcal{P}_n} \mathbf{P}_i = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$$

### Definition 3 – Relative Permutation

A permutation matrix is called **relative** if it is the ratio (or difference) of two group elements ( $i \rightarrow j$ ):  $\mathbf{P}_{ij} = \mathbf{P}_i \mathbf{P}_j^T$

## Background (2)

### Definition 4 – Permutation Synchronization Problem

Given a redundant set of measures of ratios  $P_{ij}$ :  $(i,j) \in \mathcal{E} \subset \{1, \dots, N\} \times \{1, \dots, N\}$ , where  $\mathcal{E}$  denotes the set of the edges of a directed graph of  $N$  nodes, the permutation synchronization can be formulated as the problem of recovering  $\{P_i\}$  such that the group consistency constraint is satisfied:  $P_{ij} = P_i P_j^T$ .

### Definition 5 – Doubly Stochastic (DS) Matrix

Permutation matrices are **relaxed** into their doubly-stochastic counterparts. The set of DS matrices is defined as:

$$\mathcal{DP}_n = \{\mathbf{X} \in \mathbb{R}_+^{n \times n} : \mathbf{X}\mathbf{1}_n = \mathbf{1}_n, \mathbf{1}_n^T \mathbf{X} = \mathbf{1}_n^T\}$$



# Background (3)

## Definition 6 – Birkhoff Polytope

The multinomial manifold of DS matrices is incident to the convex object called the Birkhoff Polytope. We use  $\mathcal{DP}_n$  to refer to the Birkhoff Polytope.

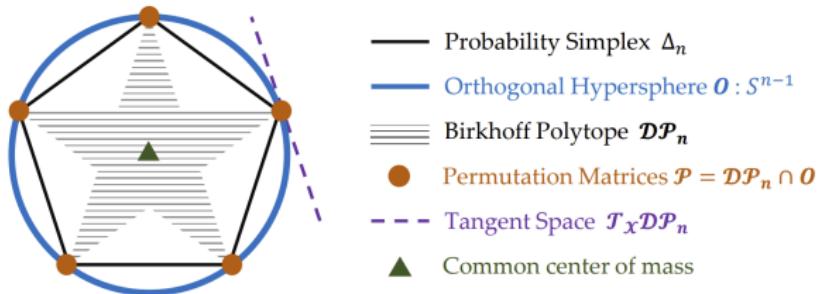


Figure 2. Simplified (matrices are vectorized) illustration of geometries we consider: (i)  $\Delta_n$  is convex, (ii)  $\mathcal{DP}_n$  is strictly contained in  $\Delta_n$ . In low dimensions, such configuration cannot exist as there is no convex shape that touches  $\Delta_n$  only on the corners.



## Background (4)

### Theorem 1 – Birkhoff-von Neumann Theorem

The convex hull of the set of all permutation matrices is the set of doubly-stochastic matrices and there exists a potentially non-unique  $\theta$  such that any DS matrix can be expressed as a linear combination of  $k$  permutation matrices:

$$\mathbf{X} = \sum_{i=1}^k \theta_i \mathbf{P}_k$$

where  $\theta_i > 0$  and  $\theta^T \mathbf{1}_k = 1$ .

Note: Finding the minimum  $k$  is NP-hard. But Marcus-Ree theorem has shown that  $k < (n - 1)^2 + 1$ .

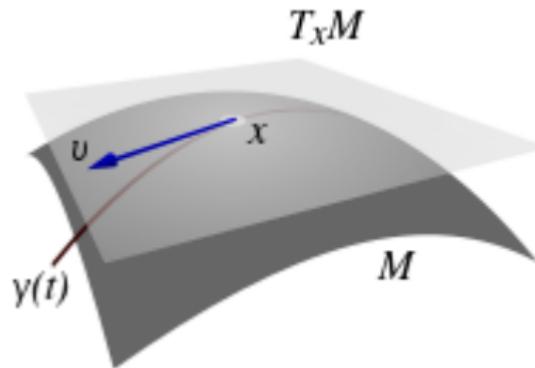
**Source code:** [https://github.com/HyTruongSon/spectral\\_permi-synch/blob/master/test\\_birkhoff.py](https://github.com/HyTruongSon/spectral_permi-synch/blob/master/test_birkhoff.py)



# Background (5)

## Notations:

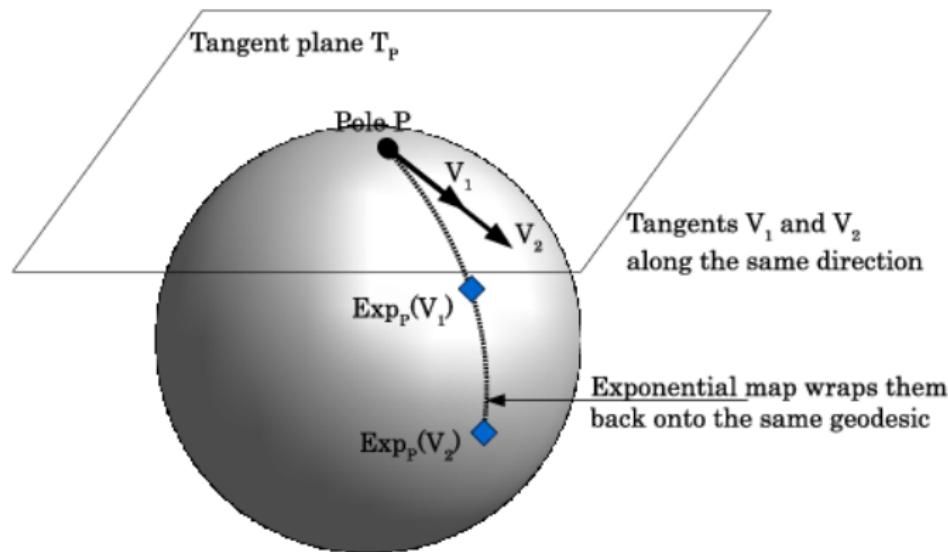
- For every point  $x$  on a given manifold  $\mathcal{M}$ , a **tangent space**  $T_x\mathcal{M}$  is defined as a first order approximation of  $\mathcal{M}$  at point  $x$ . The **exponential map**  $\exp_x(v)$  is the project of a vector  $v$  on the tangent space at point  $x$  into the corresponding point on the manifold with unit **geodesic** distance from  $x$ .



- Riemannian metric**  $g$  is a collection of inner products  $T_x\mathcal{M} \times T_x\mathcal{M} \rightarrow \mathbb{R}$ . A **geodesic** is a smooth curve  $y(t)$  which corresponds to the shortest distance between two points on a manifold.



## Background (6)



Note: I could not find better figures from the Internet with the same notation.



## Background (7)

### Theorem 2 – Projection onto the tangent space

The project operator  $\Pi_{\mathbf{X}}(\mathbf{Y})$ ,  $\mathbf{Y} \in \mathcal{DP}_n$  onto the tangent space of  $\mathbf{X} \in \mathcal{DP}_n$ , denoted as  $\mathcal{T}_{\mathbf{X}}\mathcal{DP}_n$ , is written as:

$$\Pi_{\mathbf{X}}(\mathbf{Y}) = \mathbf{Y} - (\boldsymbol{\alpha}\mathbf{1}^T + \mathbf{1}\boldsymbol{\beta}^T) \odot \mathbf{X}$$

where

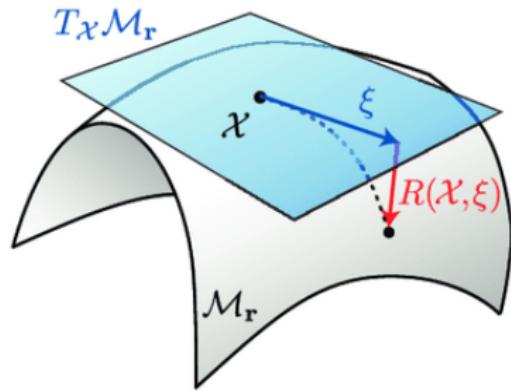
$$\boldsymbol{\alpha} = (\mathbf{I} - \mathbf{X}\mathbf{X}^T)^{\dagger}(\mathbf{Y} - \mathbf{X}\mathbf{Y}^T)\mathbf{1}$$

$$\boldsymbol{\beta} = \mathbf{Y}^T\mathbf{1} - \mathbf{X}^T\boldsymbol{\alpha}$$

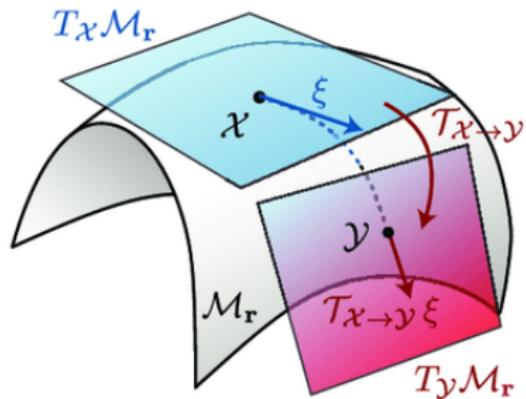
where  $\dagger$  denotes the left pseudo-inverse, and  $\odot$  denotes the Hadamard product.



# Background (8)



(a) Retraction



(b) Vector transport



## Background (9)

### Theorem 3 – First order retraction map

For a vector  $\xi x \in T_X \mathcal{DP}_n$  lying on the tangent space of  $X \in \mathcal{DP}_n$ , the first order retraction map  $R_X$  is given as follows:

$$R_X(\xi x) = \Pi(X \odot \exp(\xi x \oslash X))$$

where the operator  $\Pi$  denotes the projection onto  $\mathcal{DP}_n$  (computed by **Sinkhorn algorithm**), and  $\oslash$  is the Hadamard division.



# Background (10)

## Sinkhorn's theorem

If  $A$  is an  $n \times n$  matrix with strictly positive elements, then there exist diagonal matrices  $D$  and  $E$  with strictly positive diagonal elements such that  $DAE$  is **doubly stochastic**.

**Sinkhorn-Knopp algorithm** is to find a doubly stochastic scaling of a non-negative matrix in an iterative manner:

- ① Starting with  $D_0 = E_0 = I$ .
- ② Let  $r_k = D_{k-1}AE_{k-1}\mathbf{1}$  and assign  $D_k = \text{diag}(r_k)^{-1}$ .
- ③ Let  $c_k^T = \mathbf{1}^T D_k AE_{k-1}$  and assign  $E_k = \text{diag}(c_k)^{-1}$ .

**Reference:** A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices, Richard Sinkhorn (1964). **We will discuss this further in Part 2 of Optimal Transport series.**



## Background (11)

- In **classical optimization**: line-search methods (first-order) are based on updating the iterate by choosing a search direction and then adding a multiple of this direction to the old iterate. But it is **not** possible for general manifolds.
- The natural extension to manifolds is to follow the search direction along a path.

**Algorithm 1** (Linesearch minimization on manifolds).

**Input:**  $f : \mathcal{M} \rightarrow \mathbb{R}$ ,  $x_0 \in \mathcal{M}$ ,  $k = 0$

**repeat**

choose a descent direction  $p_k \in T_{x_k} \mathcal{M}$

choose a retraction  $R_{x_k} : T_{x_k} \mathcal{M} \rightarrow \mathcal{M}$

choose a step length  $\alpha_k \in \mathbb{R}$

set  $x_{k+1} = R_{x_k}(\alpha_k p_k)$

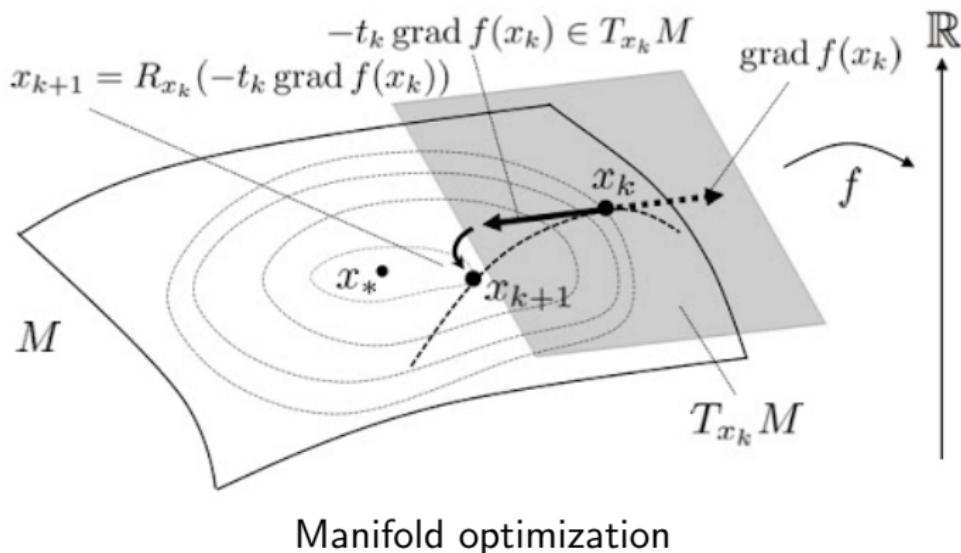
$k \leftarrow k + 1$

**until**  $x_{k+1}$  sufficiently minimizes  $f$

**Reference:** Optimization methods on Riemannian manifolds and their application to shape space, Wolfgang Ring and Benedikt Wirth.



# Background (12)



Note: I could not find a figure with the same notation with the algorithm.  
But the idea is fine.



# Markov Random Field (1)

Assume that we are provided a set of pairwise permutations  $\mathbf{P}_{ij} \in \mathcal{P}_n$  for  $(i, j) \in \mathcal{E}$ . But for our case, the graph is complete,  $\mathcal{E}$  contains all possible edges. We need to find the underlying **absolute permutations**  $\mathbf{X}_i$  for  $i \in \{1, \dots, N\}$  with respect to a common origin  $\mathbf{X}_1 = \mathbf{I}$  fixed.



Because of the lack of a manifold structure for  $\mathcal{P}_n$ , we **relax** the domain of the absolute permutations by assuming that each  $\mathbf{X}_i \in \mathcal{DP}_n$  (Birkhoff polytope). Let  $\mathbf{P} \equiv \{\mathbf{P}_{ij}\}_{(i,j) \in \mathcal{E}}$  denote the **observed** random variables. Let  $\mathbf{X} \equiv \{\mathbf{X}_i\}_{i=1}^N$  denote the **latent** variables.



## Markov Random Field (2)

The proposed **pair-wise** Markov Random Field assumes the full joint distribution factorized as follows:

$$p(\boldsymbol{P}, \boldsymbol{X}) = \frac{1}{Z} \prod_{(i,j) \in \mathcal{E}} \psi(\boldsymbol{P}_{ij}, \boldsymbol{X}_i, \boldsymbol{X}_j)$$

where  $Z$  denotes the normalization constant:

$$Z = \sum_{\boldsymbol{P} \in \mathcal{P}_n^{|\mathcal{E}|}} \int_{\mathcal{D}\mathcal{P}_n^N} \prod_{(i,j) \in \mathcal{E}} \psi(\boldsymbol{P}_{ij}, \boldsymbol{X}_i, \boldsymbol{X}_j) d\boldsymbol{X}$$

and  $\psi$  is called the **clique potential** that is defined as:

$$\psi(\boldsymbol{P}_{ij}, \boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\beta \|\boldsymbol{P}_{ij} - \boldsymbol{X}_i \boldsymbol{X}_j^T\|_F^2)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm, and  $\beta > 0$ .



## Markov Random Field (3)

If we define  $\mathbf{X}_{ij} = \mathbf{X}_i \mathbf{X}_j^T \in \mathcal{DP}_n$ , then by Birkhoff-von Neumann theorem, each  $\mathbf{X}_{ij}$  (doubly-stochastic) has a decomposition into a linear combination of permutation matrices:

$$\mathbf{X}_{ij} = \sum_{b=1}^{B_{ij}} \theta_{ij,b} \mathbf{M}_{ij,b}$$

$$\sum_{b=1}^{B_{ij}} \theta_{ij,b} = 1$$

where  $B_{ij} \in \mathbb{N}_+$ ,  $\theta_{ij,b} > 0$ , and  $\mathbf{M}_{ij,b} \in \mathcal{P}_n$ .



## Markov Random Field (4)

The pairwise MRF implies the following hierarchical decomposition:

$$p(\mathbf{X}) = \frac{1}{C} \exp \left( -\beta \sum_{(i,j) \in \mathcal{E}} \|\mathbf{X}_{ij}\|^2 \right) \prod_{(i,j) \in \mathcal{E}} Z_{ij}$$

$$p(P_{ij} | \mathbf{X}_i, \mathbf{X}_j) = \frac{1}{Z_{ij}} \exp(2\beta \text{trace}(P_{ij}^T \mathbf{X}_{ij}))$$

where  $C$  and  $Z_{ij}$  are normalization constants. For all  $i, j$ ,  $Z_{ij} \geq \prod_{b=1}^{B_{ij}} f(\beta, \theta_{ij,b})$  where  $f$  is a positive function that is increasing in both  $\beta$  and  $\theta_{ij,b}$ .



# Inference Algorithms (1)

Permutation synchronization is formulated as a probabilistic inference problem:

- ① Maximum a-posterior (MAP):

$$\mathbf{X}^* = \arg \max_{\mathbf{X} \in \mathcal{DP}_n^N} \log p(\mathbf{X} | \mathbf{P})$$

where

$$\log p(\mathbf{X} | \mathbf{P}) = -\beta \sum_{(i,j) \in \mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^T\|_F^2$$

- ② The full posterior distribution:

$$p(\mathbf{X} | \mathbf{P}) \propto p(\mathbf{X}, \mathbf{P})$$



# Inference Algorithms (2)

The MAP estimation problem can be cast as a minimization problem on  $\mathcal{DP}_n$ , given as follows:

$$\mathbf{X}^* = \arg \min_{\mathbf{X} \in \mathcal{DP}_n^N} \left\{ U(\mathbf{X}) = \sum_{(i,j) \in \mathcal{E}} \|\mathbf{P}_{ij} - \mathbf{X}_i \mathbf{X}_j^T\|_F^2 \right\}$$

where  $U$  is called the potential energy function.

Solving a difficult optimization:

- ① By using Riemannian limited-memory BFGS (LR-BFGS) →  
**Riemannian second-order method (but why we need second-order here?)**
- ② Finally, round the resulting approximate solutions into a feasible one via Hungarian algorithm (obtaining binary permutation matrices).



# Inference Algorithm (3)

---

**Algorithm 1:** Euclidean BFGS

1 **input:** A real-valued, differentiable potential energy  $U$ , initial iterate  $\mathbf{X}_0$  and initial Hessian approximation  $\mathcal{B}_0$ .

2  $k \leftarrow 0$

3 **while**  $\mathbf{x}_k$  does not sufficiently minimize  $f$  **do**

4     Compute the direction  $\boldsymbol{\eta}_k$  by solving  $\mathcal{B}_k \boldsymbol{\eta}_k = -\nabla U(\mathbf{x}_k)$  for  $\boldsymbol{\eta}_k$ .

5     Define the new iterate  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \boldsymbol{\eta}_k$ .

6     Set  $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$  and  $\mathbf{y}_k \leftarrow \nabla U(\mathbf{x}_{k+1}) - \nabla U(\mathbf{x}_k)$ .

7     Compute the new Hessian approximation:

$$\mathcal{B}_{k+1} = \mathcal{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^\top}{\mathbf{y}_k^\top \mathbf{s}_k} - \frac{\mathcal{B}_k \mathbf{s}_k \mathbf{s}_k^\top \mathcal{B}_k}{\mathbf{s}_k^\top \mathcal{B}_k \mathbf{s}_k}. \quad (19)$$

$k \leftarrow k + 1$ .

---

## Euclidean second-order method

BFGS algorithm belongs to the group of Quasi-Newton methods in which the Hessian matrix of second derivatives is **not** computed. Instead, the Hessian matrix is approximated using updates specified by gradient evaluations (or approximate gradient evaluations).



# Inference Algorithm (4)

---

**Algorithm 2:** General Riemannian Line Search Minimizer

---

- 1 **input:** A Riemannian manifold  $\mathcal{M}$ , a retraction operator  $R$  and initial iterate  $\mathbf{x}_k \in \mathcal{M}$  where  $k = 0$ .
- 2 **while**  $\mathbf{x}_k$  does not sufficiently minimize  $f$  **do**
- 3     Pick a gradient related descent direction  $\boldsymbol{\eta}_k \in T_{\mathbf{x}_k} \mathcal{M}$ .
- 4     Choose a retraction  $R_{\mathbf{x}_k} : T_{\mathbf{x}_k} \mathcal{M} \rightarrow \mathcal{M}$ .
- 5     Choose a step length  $\tau_k \in \mathbb{R}$ .
- 6     Set  $\mathbf{x}_{k+1} \leftarrow R_{\mathbf{x}_k}(\tau_k \boldsymbol{\eta}_k)$ .
- 7      $k \leftarrow k + 1$ .

---

---

**Algorithm 3:** General Riemannian Steepest Descent with Armijo Line Search

---

- 1 **input:** A Riemannian manifold  $\mathcal{M}$ , a retraction operator  $R$ , the projection operator onto the tangent space  $\Pi_{\mathbf{x}_k} : \mathbb{R}^n \rightarrow T_{\mathbf{x}_k} \mathcal{M}$ , a real-valued, differentiable potential energy  $f$ , initial iterate  $\mathbf{x}_0 \in \mathcal{M}$  and the Armijo line search scalars including  $c$ .
- 2 **while**  $\mathbf{x}_k$  does not sufficiently minimize  $f$  **do**
- 3     // Euclidean gradient to Riemannian direction
- 4      $\boldsymbol{\eta}_k \leftarrow -\text{grad}f(\mathbf{x}_k) \triangleq \Pi_{\mathbf{x}_k}(-\nabla f(\mathbf{x}_k))$ .
- 5     Select  $\mathbf{x}_{k+1}$  such that:  
$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq c(f(\mathbf{x}_k) - f(R_{\mathbf{x}}(\tau_k \boldsymbol{\eta}_k))), \quad (22)$$
- 6     where  $\tau_k$  is the Armijo step size.
- 7      $k \leftarrow k + 1$ .

---

# Inference Algorithm (5)

If we don't want to solve the optimization problem by first/second-order methods, there is an alternative – **Riemannian Langevin Monte Carlo** (this can be understood as a first-order method):

$$\mathbf{v}_i^{(k+1)} = \nabla_{\mathbf{x}_i} U(\mathbf{X}_i^{(k)}) + \sqrt{2h/\beta} \mathbf{Z}_i^{(k+1)}$$

$$\mathbf{X}_i^{(k+1)} = R_{\mathbf{X}_i^{(k)}}(\mathbf{v}_i^{(k+1)})$$

for  $\forall i \in \{1, \dots, N\}$  and  $h > 0$  denotes the step-size,  $k$  denotes the iterations,  $Z_i^{(k)}$  denotes standard Gaussian random variables in  $\mathbb{R}^{n \times n}$ , and  $\mathbf{X}_i^{(0)}$  is the initial absolute permutations.

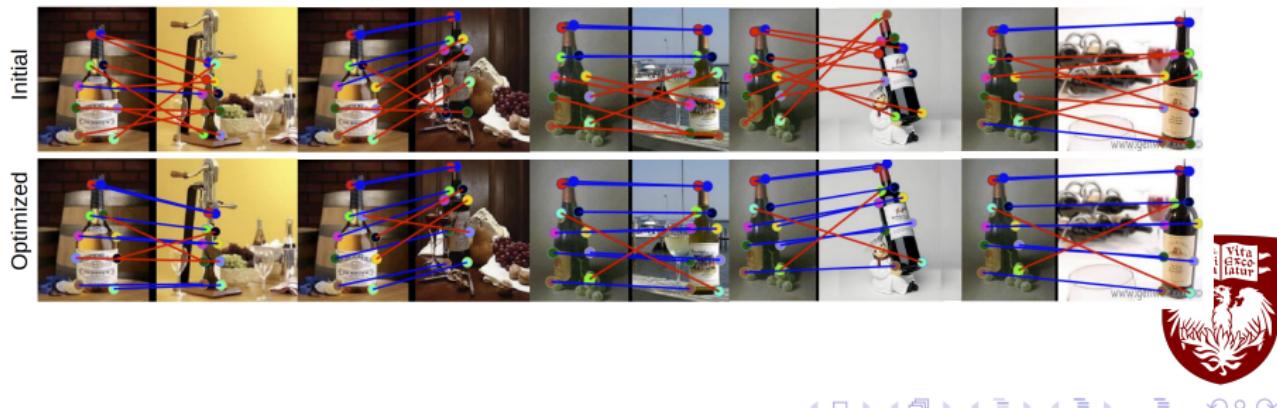


# 2D multi-image matching

The standard evaluation metric:

$$R(\{\hat{\mathbf{P}}_i\} | \mathbf{P}^{\text{gnd}}) = \frac{1}{n|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} \mathbf{P}_{ij}^{\text{gnd}} \odot (\hat{\mathbf{P}}_i \hat{\mathbf{P}}_j^T)$$

where  $\mathbf{P}_{ij}^{\text{gnd}}$  are the ground-truth relative transformations, and  $\hat{\mathbf{P}}_i$  is an estimated permutation.  $R = 0$ : nothing correct.  $R = 1$ : perfect.



# 3D correspondences

Initial correspondences are obtained by running siamese **PointNet** (another paper of Guibas' team <http://stanford.edu/~rqi/pointnet/>).

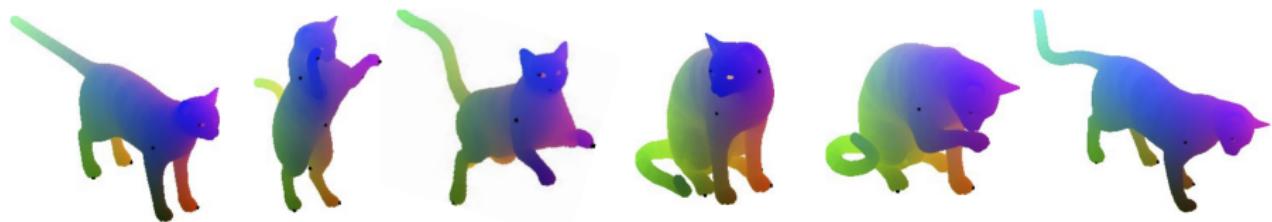


Figure 10. Visualizations of the *cat* object from the Tosca dataset with the ground truth correspondences depicted.

