

# Detecting Approximate Reflection Symmetry in a Point Set using Optimization on Manifold

Rajendra Nagar\* and Shanmuganathan Raman†

Electrical Engineering, Indian Institute of Technology Gandhinagar, India, 382355

Email: \*rajendra.nagar@iitgn.ac.in, †shanmuga@iitgn.ac.in

**Abstract**—We propose an algorithm to detect approximate reflection symmetry present in a set of volumetrically distributed points belonging to  $\mathbb{R}^d$  containing a distorted reflection symmetry pattern. We pose the problem of detecting approximate reflection symmetry as the problem of establishing correspondences between the points which are reflections of each other and we determine the reflection symmetry transformation. We formulate an optimization framework in which the problem of establishing the correspondences amounts to solving a linear assignment problem and the problem of determining the reflection symmetry transformation amounts to solving an optimization problem on a smooth Riemannian product manifold. The proposed approach estimates the symmetry from the geometry of the points and is descriptor independent. We evaluate the performance of the proposed approach on the standard benchmark dataset and achieve the state-of-the-art performance. We further show the robustness of our approach by varying the amount of distortion in a perfect reflection symmetry pattern where we perturb each point by a different amount of perturbation. We demonstrate the effectiveness of the method by applying it to the problem of 2-D and 3-D reflection symmetry detection along with comparisons.

**Index Terms**—Symmetry, Optimization, Manifolds.

## I. INTRODUCTION

**S**YMMETRY present in natural and man-made objects enriches the objects to be physically balanced, beautiful, easy to recognize, and easy to understand. Characterizing and finding the symmetry has been an active topic of research in computer vision and computer graphics as physical objects form the basis for these research areas. The digitized objects are mainly represented in the form of meshes, volumes, sets of points, and images. The primary objective has been to detect symmetry in objects depicted through these different representations. We particularly aim to detect reflection symmetry present in objects represented by a set of finite number of points belonging to  $\mathbb{R}^d$ . In Fig. 1, we present an example result of the proposed approach for illustration.

The motivation behind detecting symmetry in higher dimensional spaces ( $d > 3$ ) is inspired by the fact that many physical data points reside in the space of dimensions greater than three. For example, an RGB-D image captured using a Kinect sensor, which has become a major tool for interaction of human with machine, has four dimensions at each pixel location. Another example is the embedding of feature points or shapes into a higher-dimensional space. In the scale invariant feature transform (SIFT) algorithm, each keypoint is represented in a 128-dimensional space [1]. We not only target data residing in 2-D (image) and 3-D (point cloud), but also develop a generic framework to detect symmetry in higher dimensional data.

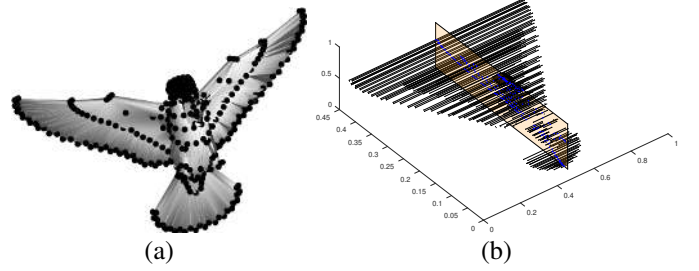


Fig. 1. Established correspondences (shown in (b)) between the reflection symmetry points sampled from the input model (shown in (a)) using the proposed approach.

The problem of establishing correspondences between reflection symmetry points and determining the hyperplane of reflection symmetry has been extensively studied due to its astounding applications such as compression of objects, symmetrization, shape matching, and symmetry aware segmentation of shapes [2]. Most of the existing algorithms attempt this problem by using surface signatures such as Gaussian curvature, eigenbases of the Laplace-Beltrami operator, and heat kernels for the points sampled on a given surface ([3], [4], [2]). The challenge we face is that, we only have a set of discrete points in  $\mathbb{R}^d$ . We can not take benefits from local surface signatures by fitting a surface over these points. For the case  $d = 2$ , an explanation could be that the prominent surface signatures, such as Gaussian curvatures, are meaningful only if the surface is non-linear. For the case  $d \geq 3$ , an explanation could be that if the point set represents a volumetric shape, fitting a surface could be hard and eigenbases of Laplace-Beltrami operator are not defined for a set of finite points since it is not a compact manifold without the boundary [5]. Prominent methods such as [6] and [7] are independent of surface features and employ randomized algorithms to establish correspondences between the reflection symmetry points. However, they require fine tuning of a hyper-parameter to handle the reflection symmetry patterns perturbed by an unknown source of noise and an improper choice of this parameter could lead to higher time complexity.

Both these categories of algorithms are sequential in the sense that they first establish the correspondences between the reflection symmetry points and then determine the reflection symmetry hyperplane. Therefore, many outlier correspondences could be detected along with the correct correspondences. In summary, detecting symmetry in a set containing a finite number of points is a non-trivial problem. In this

work, we propose an optimization framework where we jointly establish correspondences between reflection symmetry points and determine the reflection symmetry hyperplane in a set of points containing a distorted reflection symmetry pattern. In order to design the cost function, we introduce an affine transformation to obtain the reflection point of a point in  $\mathbb{R}^d$ . The main intuition behind forming this cost function is that the reflection point of a point obtained through the optimal reflection hyperplane should be present closest to its ground-truth reflection point.

The primary contributions of this work are listed below.

- 1) We propose an optimization based algorithm to establish correspondences between the reflection symmetry points and determine the reflection symmetry transformation in a set of discrete points residing in  $\mathbb{R}^d$  containing a distorted reflection symmetry pattern.
- 2) We show that the proposed optimization framework is convex in translation and correspondences matrix, and *locally convex* in each of the rotation matrices.
- 3) The proposed approach is shown to not use any shape descriptors and can be applied to point sets obtained by sampling any shape residing in  $\mathbb{R}^d$ .
- 4) We demonstrate the effectiveness of the proposed approach by detecting symmetry in 2-D images and 3-D point clouds.

We organize the remainder of the paper as follows. In §II, we present the related works to our approach. In §III-A, we formulate the energy minimization problem. In §III-B, we find the optimal rotations and translation. In §III-C, we find the optimal mirror symmetric correspondences. In §III-D, we prove the convergence properties. In §IV, we report the computational complexity of our algorithm. In §V, we report the results and the evaluation of the proposed approach. In §VI, we conclude the work with future directions.

## II. RELATED WORKS

The problem of characterizing and detecting the reflection symmetry in digitized objects has been extensively studied. The works [8] and [2] provide a survey of the symmetry detection algorithms. The symmetry detection algorithms can be categorized based on either the form of the input data or whether the algorithm is dependent or independent of the surface features. General forms of the input data are: set of points, mesh, volume, and image. Most of the methods for symmetry detection in meshes first extract salient keypoints on the surface and then describe each point using local surface features. The prominent surface features are: Gaussian curvatures, slippage features, moments, geodesic distances, and extended Gaussian images ([8], [2]).

### **Symmetry detection in a set of points without features.**

These algorithms detect reflection symmetry in a set of points without using surface features. Our work also falls in this category. In the work by Zabrodsky *et al.*, the authors find the closest shape to a given shape represented by a set of points in  $\mathbb{R}^2$  and it requires point correspondences [9]. However, our goal is different in the sense that we find reflection correspondences within the given set of points in  $\mathbb{R}^d$ . In the

work by Lipman *et al.*, the authors propose the concept of symmetry factored embedding where they represent pairs of points which are in the same orbit in a new space and propose the concept of symmetry factored distance to find the distance between such pairs [6]. In the work by Xu *et al.*, the authors detect multi-scale symmetry [7]. The authors use a randomized algorithm to detect the correspondences efficiently. However, performance degrades as the perfect pattern gets perturbed due to noisy measurements. We compare the correspondences established by our method to that of this method and show that our method performs better than this method when the patterns are perturbed. It is fair to compare with this method on the perturbed patterns because most of the real world patterns are not perfectly symmetric, e.g., human face and butterfly wings. In the works by Combès *et al.* [10], Speciale *et al.* [11], Ecins *et al.* [12], Cicconet *et al.* [13], Li *et al.* [14], and Sipiran *et al.* [15], the authors automatically detect the symmetry plane in a point cloud. But, the methods in [10], [11], [13], [14], and [15] do not establish correspondences. However, correspondences are an important aspect as shown in ([7], [6]). Ecins *et al.* proposed an ICP based approach [12] where they used the normals at each point to determine the symmetry. However, this method is applicable only to non-volumetric point clouds, i.e., points sampled from a surface.

### **Symmetry detection in meshes using surface features.**

These algorithms either directly use surface patches described using local features or first detect the salient keypoints on the surface and describe them using the local surface features. Here, we review only the salient works to give an idea of these algorithms. Mitra *et al.* detect partial and approximate symmetries in 3D models [4]. They start with sampling salient keypoints on the surface and pair them up using their local principal curvatures. Then using the Hough transformation, they find the pairs of reflection symmetry points. Then in the Hough transformation space, they perform the clustering of the pairs to determine all the partial symmetries. Martinet *et al.* detect symmetries by generalized moment functions where the shape symmetry gets inherited as symmetry in these functions [16]. Raviv *et al.* detect symmetry in non-rigid shapes by observing that the intrinsic geometry of a shape is invariant under non-rigid shape transformations [17]. Berner *et al.* start with constructing a graph based on the similarity of slippage features detected on the surface [3]. Then, they detect the structural regularities by matching the sub-graphs. Cohen *et al.* detect symmetry using geometric and image cues [18]. They use it to reconstruct accurate 3D models. We refer the reader to some of the pioneering works for more details on this category ([19], [20], [21], [22], [23], [24], [25], [26], [27]). There exist algorithms which find symmetry in meshes and volume without sampling keypoints. The works described in ([28], [29], [5], [30], [31], [32], [33], [34], [35]) belong to this category.

**Symmetry detection algorithm for real images.** These algorithms primarily rely on the local image features such as edge orientations, curvatures, and gradients. The recent works such as ([36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47]) present excellent algorithms for reflection symmetry detection in images. Given the accurate detection of

keypoints, the algorithm developed in this work can be used to detect reflection symmetry in images without using local features.

Our algorithm is similar to Iterative Closest Point (ICP) algorithm ([48], [49]) only in the sense that we also follow the alternation between the optimization of reflection transformation (rotation and translation in ICP) and the correspondences between the mirror symmetric points (correspondences between the points of two different shapes in ICP). Our algorithm differs from the ICP algorithm since ICP has a different error function in the transformation parameters than the error function of our problem. Furthermore, our matching is bijective since we impose the bijectivity constraints in our optimization framework. These constraints ensure that each point has exactly one mirror image point.

### III. PROPOSED APPROACH

Consider a set  $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$  of points, where  $\mathbf{x}_i \in \mathbb{R}^d$ , containing a distorted reflection symmetry pattern. Our goal is to determine the reflection symmetry transformation and establish the correspondences between the reflection symmetry points. In Fig. 2, we show the graphical representation of our problem. We formulate an optimization framework in which both the correspondences between reflection symmetry points and the reflection symmetry transformation are variables as described below. We use the notation  $[k]$  for the set  $\{1, 2, \dots, k\}$ , where  $k$  is a natural number.

#### A. Problem Formulation

We introduce reflection transformation in  $\mathbb{R}^d$  in order to obtain the reflection of a point through a hyperplane  $\pi$ , not necessarily passing through the origin. The intuition is based on the fact that any hyperplane in  $\mathbb{R}^d$  is a  $d - 1$  dimensional subspace. Therefore, it can be made parallel to the subspace spanned by any  $d - 1$  coordinate axes by translating the origin of the coordinate system on the hyperplane  $\pi$  and then rotating these  $d - 1$  axes sequentially (by the angle between the hyperplane  $\pi$  and the axis). In this new coordinate system, the reflection of a point through the hyperplane  $\pi$  can be obtained by multiplying the coordinate corresponding to the remaining axis of the point by  $-1$ . Then the reflection in the original coordinate system is obtained by applying the inverse procedure.

**Definition 1.** *The reflection point  $\mathbf{x}_{i'}$   $\in \mathbb{R}^d$  of a point  $\mathbf{x}_i \in \mathbb{R}^d$  through the reflection symmetry hyperplane  $\pi$  is determined by an affine transformation as shown in Equation 1.*

$$\mathbf{x}_{i'} = \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top (\mathbf{x}_i - \mathbf{t}) + \mathbf{t}. \quad (1)$$

Here,  $i, i' \in [n]$ ,  $\mathbf{t} \in \mathbb{R}^d$  is the translation vector which translates the origin of the coordinate system on the hyperplane  $\pi$ ,  $\mathbf{R}_u$  is a rotation matrix of size  $d \times d$  that rotates the  $u^{\text{th}}$  axis about the origin such that it becomes perpendicular to the normal of the hyperplane  $\pi$ , and the matrix  $\mathbf{E}$  is defined as  $\mathbf{E} = \begin{bmatrix} \mathbf{I}_{d-1} & \mathbf{0}_{d-1} \\ \mathbf{0}_{d-1}^\top & -1 \end{bmatrix}$  and satisfies  $\mathbf{E}^\top = \mathbf{E}$ ,  $\mathbf{E}^\top \mathbf{E} = \mathbf{I}_d$ . The

matrix  $\mathbf{R}_u$  is an orthogonal matrix ( $\mathbf{R}_u^\top \mathbf{R}_u = \mathbf{R}_u \mathbf{R}_u^\top = \mathbf{I}_d$ ) with determinant equal to  $+1$ ,  $\forall u \in \{1, 2, \dots, d - 1\}$ . Here,  $\mathbf{0}_{d-1}$  is a vector of size  $(d - 1) \times 1$  with all the coordinates equal to zero and  $\mathbf{I}_{d-1}$  is an identity matrix of size  $(d - 1) \times (d - 1)$ .

Now, we introduce the essential properties of this transformation in order to formulate the problem. We show that the rotation matrices ( $\mathbf{R}_1, \dots, \mathbf{R}_{d-1}$ ) and the translation vector  $\mathbf{t}$  uniquely determine the reflection hyper-plane  $\pi$ . We let  $\mathbf{T} = \prod_{u=1}^{d-1} \mathbf{R}_u$  throughout this paper and note that it is again an orthogonal matrix with determinant equal to  $+1$ .

**Theorem 1:** *The point  $\mathbf{x}_{i'}$  is the reflection of the point  $\mathbf{x}_i$  through the hyperplane  $\pi$  if and only if the point  $\mathbf{x}_i$  is the reflection of the point  $\mathbf{x}_{i'}$  through the hyperplane  $\pi$ .*

**Proof:** We prove the forward direction of the Theorem 1, since the backward direction can be proved in a similar way. Let us assume that the point  $\mathbf{x}_{i'}$  is the reflection of the point  $\mathbf{x}_i$ . Therefore, Equation (1) holds true. Now, we multiply Equation (1) by  $\mathbf{TET}^\top$  from left and use the identities  $\mathbf{E}^\top = \mathbf{E}$ ,  $\mathbf{EE} = \mathbf{I}_d$ ,  $\mathbf{T}^\top \mathbf{T} = \mathbf{TT}^\top = \mathbf{I}_d$  to achieve,

$$\begin{aligned} \mathbf{TET}^\top \mathbf{x}_{i'} &= \mathbf{x}_i - \mathbf{t} + \mathbf{TET}^\top \mathbf{t} \\ \Rightarrow \mathbf{x}_i &= \mathbf{TET}^\top (\mathbf{x}_{i'} - \mathbf{t}) + \mathbf{t}. \end{aligned} \quad (2)$$

□

**Theorem 2:** *The normal vector of the reflection hyper-plane  $\pi$  lies in the null space of the matrix  $\mathbf{I}_d + \mathbf{TET}^\top$ , the hyper-plane  $\pi$  passes through  $\mathbf{t}$ , and the null space of the matrix  $\mathbf{I}_d + \mathbf{TET}^\top$  is an one-dimensional subspace of  $\mathbb{R}^d$ .*

**Proof:** We subtract Equation (1) from Equation (2) to achieve  $\mathbf{x}_i - \mathbf{x}_{i'} = \mathbf{TET}^\top (\mathbf{x}_{i'} - \mathbf{x}_i) \Rightarrow (\mathbf{I}_d + \mathbf{TET}^\top) (\mathbf{x}_i - \mathbf{x}_{i'}) = \mathbf{0}$ .

Therefore, the normal to the reflection hyperplane  $\pi$ , which is in the direction of the vector  $(\mathbf{x}_i - \mathbf{x}_{i'})$ , lies in the null space of the matrix  $(\mathbf{I}_d + \mathbf{TET}^\top)$ . It is easy to show that the reflection hyperplane  $\pi$  passes through the translation  $\mathbf{t}$  by noting that the reflection point of the point  $\mathbf{t}$  is  $\mathbf{t}$ . This is possible only if the point  $\mathbf{t}$  lies on the reflection hyperplane  $\pi$ . We prove that the null space of the matrix  $\mathbf{I}_d + \mathbf{TET}^\top$  is an one-dimensional subspace of  $\mathbb{R}^d$  in order to show that there exists a unique hyperplane  $\pi$ . The nullspace of a matrix is the space spanned by the eigenvectors corresponding to the zero eigenvalue. Let  $\mathbf{p} = [p_1 \ p_2 \ \dots \ p_d]^\top \in \mathbb{R}^d$  be any vector. If  $\mathbf{p}$  is an eigenvector corresponding to the zero eigenvalue of the matrix  $\mathbf{I}_d + \mathbf{TET}^\top$ , then we must have

$$\begin{aligned} \mathbf{p}^\top (\mathbf{I}_d + \mathbf{TET}^\top) \mathbf{p} = 0 &\Rightarrow \mathbf{p}^\top \mathbf{I}_d \mathbf{p} + (\mathbf{T}^\top \mathbf{p})^\top \mathbf{E} (\mathbf{T}^\top \mathbf{p}) = 0 \\ \Rightarrow \mathbf{p}^\top \mathbf{I}_d \mathbf{p} + \mathbf{b}^\top \mathbf{E} \mathbf{b} = 0 &\Rightarrow \sum_{u=1}^d p_u^2 + \sum_{u=1}^{d-1} b_u^2 - b_d^2 = 0 \\ \Rightarrow \sum_{u=1}^d p_u^2 + \sum_{u=1}^d b_u^2 - 2b_d^2 = 0. & \end{aligned} \quad (3)$$

Here,  $\mathbf{b} = \mathbf{T}^\top \mathbf{p}$ . We note that  $\|\mathbf{b}\|_2^2 = (\mathbf{T}^\top \mathbf{p})^\top (\mathbf{T}^\top \mathbf{p}) = \mathbf{p}^\top \mathbf{p} = \|\mathbf{p}\|_2^2$ . Therefore, from Equation (3) we have

$$\sum_{u=1}^d p_u^2 + \sum_{u=1}^d p_u^2 - 2b_d^2 = 0 \Rightarrow \sum_{u=1}^d p_u^2 = b_d^2 \Rightarrow \sum_{u=1}^{d-1} b_u^2 = 0.$$

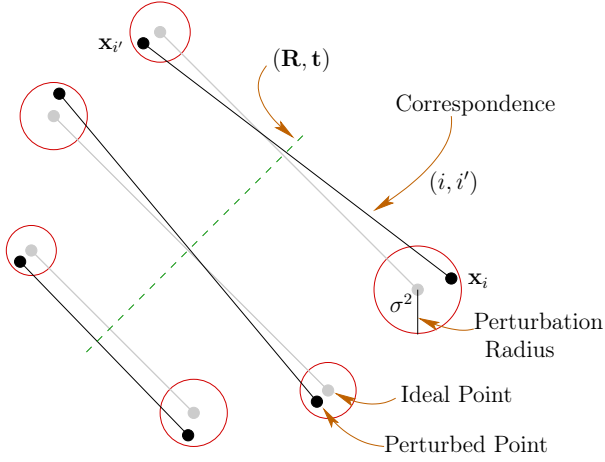


Fig. 2. Each point of a perfect pattern, shown in gray color, is perturbed to a point within a circular region around it where the radius is different for all the points and is unknown. Our goal is to determine the correspondences  $(i, i')$  and the reflection transformation  $(\mathbf{R}, \mathbf{t})$ .

Therefore,  $b_1 = b_2 = \dots = b_{d-1} = 0$  and  $b_d \in \mathbb{R}$ . Hence, the vector  $\mathbf{b}$  lies in the one dimensional space  $\{\mathbf{q} : q_1 = q_2 = \dots = q_{d-1} = 0, q_d \in \mathbb{R}\}$ . Since  $\mathbf{b} = \mathbf{T}^\top \mathbf{p} \Rightarrow \mathbf{p} = \mathbf{T}\mathbf{b}$ . Since the rotation does not change the dimension of a linear space, the vector  $\mathbf{p}$  also lies in one dimensional space.  $\square$

Given the set  $\mathcal{S}$ , our goal is to find all the correct reflection correspondences  $(i, i') \in [n] \times [n]$  and the matrices  $(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{d-1}, \mathbf{t})$  which define the reflection symmetry hyperplane  $\pi$ . We represent all the correspondences by a permutation matrix  $\mathbf{P} \in \{0, 1\}^{n \times n}$ , such that  $\mathbf{P}_{ii'} = 1$  if the point  $\mathbf{x}_{i'}$  is the reflection point of the point  $\mathbf{x}_i$  and  $\mathbf{P}_{ii'} = 0$ , otherwise. Here, we note from Theorem 1 that  $\mathbf{P}_{ii'} = 1 \Leftrightarrow \mathbf{P}_{i'i} = 1$ .

Now, we let  $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{d-1}) \in \mathbb{V}$ . Here,  $\mathbb{V} = \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times d} \times \dots \times \mathbb{R}^{d \times d}$ . Let  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  be the matrix containing all the points of the set  $\mathcal{S}$  as its columns. Since the  $i^{\text{th}}$  column of the matrix  $\mathbf{X}\mathbf{P}$  is the reflection point of the point  $\mathbf{x}_i$ , the reflection transformation  $(\mathbf{R}, \mathbf{t})$  maps the matrix  $\mathbf{X}$  to the reflected points matrix  $\mathbf{X}\mathbf{P}$ . Using Equation 1, we write the reflected points in the form of the matrix  $\mathbf{T}\mathbf{E}\mathbf{T}^\top(\mathbf{X} - \mathbf{t}\mathbf{e}^\top) + \mathbf{t}\mathbf{e}^\top$ , where  $\mathbf{e} = [1 \ 1 \ \dots \ 1]^\top$  is a vector of size  $n \times 1$ . Therefore, Equation (4) holds true when the input set contains a perfect reflection symmetry pattern.

$$\mathbf{T}\mathbf{E}\mathbf{T}^\top(\mathbf{X} - \mathbf{t}\mathbf{e}^\top) + \mathbf{t}\mathbf{e}^\top = \mathbf{X}\mathbf{P}. \quad (4)$$

In practice, a reflection symmetry pattern might have been distorted. Therefore, we would be able to find only the *approximate reflection symmetry*. We find the reflection transformation  $(\mathbf{R}, \mathbf{t})$  and the correspondences matrix  $\mathbf{P}$  in such a way that the symmetry error, which we define as  $\|\mathbf{T}\mathbf{E}\mathbf{T}^\top(\mathbf{X} - \mathbf{t}\mathbf{e}^\top) + \mathbf{t}\mathbf{e}^\top - \mathbf{X}\mathbf{P}\|_{\mathbb{F}}^2$ , is minimized. Here  $\|\cdot\|_{\mathbb{F}}$  is the Frobenius norm operator. We frame this problem in an optimization

framework as shown in Equation (5).

$$\begin{aligned} \min_{\mathbf{R}, \mathbf{t}, \mathbf{P}} \quad & \left\| \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top (\mathbf{X} - \mathbf{t}\mathbf{e}^\top) + \mathbf{t}\mathbf{e}^\top - \mathbf{X}\mathbf{P} \right\|_{\mathbb{F}}^2 \\ \text{s.t.} \quad & \mathbf{P}\mathbf{e} = \mathbf{e}, \mathbf{P}^\top \mathbf{e} = \mathbf{e}, \mathbf{P} \in \{0, 1\}^{n \times n}, \\ & \mathbf{R}_u^\top \mathbf{R}_u = \mathbf{I}_d = \mathbf{R}_u \mathbf{R}_u^\top, \det(\mathbf{R}_u) = 1, \mathbf{R}_u \in \mathbb{R}^{d \times d}, \\ & \forall u \in [d-1], \mathbf{t} \in \mathbb{R}^d. \end{aligned} \quad (5)$$

By imposing the constraints  $\mathbf{P}\mathbf{e} = \mathbf{e}$  and  $\mathbf{P}^\top \mathbf{e} = \mathbf{e}$ , we ensure that each point has only one reflection point. We adopt an alternating optimization approach to solve the problem defined in Equation (5). We start with initializing the reflection transformation  $(\mathbf{R}, \mathbf{t})$  and solve for the optimal correspondences  $\mathbf{P}$  and then for this optimal  $\mathbf{P}$ , we solve for optimal the  $(\mathbf{R}, \mathbf{t})$ . We continue to alternate till convergence.

Once  $\mathbf{P}$  is fixed, if we minimize the cost over the set  $\mathbb{V} \times \mathbb{R}^d$ , then we have to make sure that the orthogonality and the unit determinant constraints hold true for the matrices  $\mathbf{R}_u, \forall u \in [d-1]$ . One approach could be the Lagrange augmentation which requires us to handle  $3d - 3$  additional Lagrange multipliers. However, we observe that the set  $\mathcal{M} = \{(\mathbf{R}_1, \dots, \mathbf{R}_{d-1}, \mathbf{t}) : \mathbf{R}_u^\top \mathbf{R}_u = \mathbf{R}_u \mathbf{R}_u^\top = \mathbf{I}_d, \det(\mathbf{R}_u) = 1, \mathbf{R}_u \in \mathbb{R}^{d \times d}, \forall u \in [d-1], \mathbf{t} \in \mathbb{R}^d\}$  of constraints is a *smooth Riemannian product manifold* over which the optimization algorithms are well studied [50].

We solve the sub-optimization problem for optimal  $(\mathbf{R}, \mathbf{t})$  on a manifold which we discuss in Section III-B. We observe that the optimization of Equation (5) for  $\mathbf{P}$  is a standard *linear assignment problem* for which we formulate an integer linear program which we discuss in Section III-C.

### B. Optimizing reflection transformation $(\mathbf{R}, \mathbf{t})$

In this step, we fix the correspondences matrix  $\mathbf{P}$  and find the optimal reflection transformation  $(\mathbf{R}, \mathbf{t})$  by taking advantages from the *differential structure* of the set  $\mathcal{M}$ . We shall now briefly introduce the differential geometry of the set  $\mathcal{M}$ .

**Differential geometry of the set  $\mathcal{M}$  of constraints.** In order to introduce the essential differential geometry of the set  $\mathcal{M}$ , we follow [50]. The elements of the set  $\mathcal{M}$  are of the form  $(\mathbf{R}, \mathbf{t}) \simeq (\mathbf{R}_1, \dots, \mathbf{R}_{d-1}, \mathbf{t})$ . All the orthogonal matrices (each for rotation along a single axis) with determinant +1 form a Lie group, also known as *special orthogonal group*, which is a smooth Riemannian manifold. The Euclidean space  $\mathbb{R}^d$  is also a smooth Riemannian manifold. Therefore the set  $\mathcal{M}$  is a product manifold,  $\mathcal{SO}(2, d) \times \dots \times \mathcal{SO}(2, d) \times \mathbb{R}^d$ , the product of  $d - 1$  special orthogonal groups  $\mathcal{SO}(2, d)$  and an Euclidean space  $\mathbb{R}^d$ . Each rotation matrix performs rotation about a single axis. Therefore, all the possible rotation matrices about a particular axis form a  $\mathcal{SO}(2)$  embedded in the Euclidean space  $\mathbb{R}^{d \times d}$ . We denote this group as  $\mathcal{SO}(2, d)$ .

The tangent space  $\mathcal{T}_{(\mathbf{R}, \mathbf{t})}\mathcal{M}$  at the point  $(\mathbf{R}, \mathbf{t}) \in \mathcal{M}$  is

$$\{(\mathbf{R}\mathbf{\Omega}, \mathbf{t}) : \mathbf{\Omega}_u^\top = -\mathbf{\Omega}_u, \mathbf{\Omega}_u \in \mathbb{R}^{d \times d}, \forall u \in [d-1], \mathbf{t} \in \mathbb{R}^d\}. \quad (6)$$

Here,  $\mathbf{R}\mathbf{\Omega} = (\mathbf{R}_1\mathbf{\Omega}_1, \dots, \mathbf{R}_{d-1}\mathbf{\Omega}_{d-1})$ . The Riemannian metric  $\langle \cdot, \cdot \rangle_{(\mathbf{R}, \mathbf{t})}$  on the product manifold  $\mathcal{M}$ , which gives the intrinsic distance between two elements  $(\mathbf{R}\mathbf{\Omega}, \mathbf{\eta}_t)$  and

$(\mathbf{R}\Omega', \eta'_t)$  of the tangent space at the point  $(\mathbf{R}, \mathbf{t})$  of the manifold  $\mathcal{M}$ , is defined in Equation (7).

$$\langle (\mathbf{R}\Omega, \eta_t), (\mathbf{R}\Omega', \eta'_t) \rangle_{(\mathbf{R}, \mathbf{t})} = \eta_t^\top \eta'_t + \sum_{u=1}^{d-1} \text{trace}(\Omega_u^\top \Omega'_u). \quad (7)$$

Let  $\bar{f} : \mathbb{V} \times \mathbb{R}^d \rightarrow \mathbb{R}$  be a scalar function. Let the function  $f = \bar{f}|_{\mathcal{M}}$  be the restriction of the function  $\bar{f}$  on the product manifold  $\mathcal{M}$ . Since the product manifold  $\mathcal{M}$  is a *submanifold* of the Riemannian manifold  $\mathbb{V} \times \mathbb{R}^d$ , the Riemannian gradient of the function  $f$  at the point  $(\mathbf{R}, \mathbf{t})$  is obtained by projecting the Riemannian gradient of the function  $\bar{f}$  at the point  $(\mathbf{R}, \mathbf{t}) \in \mathbb{V} \times \mathbb{R}^d$  on the tangent space at the point  $(\mathbf{R}, \mathbf{t}) \in \mathcal{M}$ . Therefore, the Riemannian gradient of the function  $f$  at the point  $(\mathbf{R}, \mathbf{t})$  is defined in Equation (8).

$$\text{grad } f(\mathbf{R}, \mathbf{t}) = (\mathbb{P}_{\mathbf{R}}(\nabla_{\mathbf{R}} \bar{f}), \mathbb{P}_{\mathbf{t}}(\nabla_{\mathbf{t}} \bar{f})) \in \mathcal{T}_{(\mathbf{R}, \mathbf{t})} \mathcal{M}. \quad (8)$$

Since the tangent space at a point in an Euclidean space is again an Euclidean space, the second component is given by  $\mathbb{P}_{\mathbf{t}}(\nabla_{\mathbf{t}} \bar{f}) = \nabla_{\mathbf{t}} \bar{f}$ . The first component is defined as

$$\mathbb{P}_{\mathbf{R}}(\nabla_{\mathbf{R}} \bar{f}) = (\mathbb{P}_{\mathbf{R}_1}(\nabla_{\mathbf{R}_1} \bar{f}), \dots, \mathbb{P}_{\mathbf{R}_{d-1}}(\nabla_{\mathbf{R}_{d-1}} \bar{f})).$$

Here,

$$\mathbb{P}_{\mathbf{R}_j}(\nabla_{\mathbf{R}_j} \bar{f}) = \mathbf{R}_j \text{skew}(\mathbf{R}_j^\top \nabla_{\mathbf{R}_j} \bar{f}),$$

where  $\text{skew}(\mathbf{A}) = 0.5(\mathbf{A} - \mathbf{A}^\top)$ . We define  $\xi_{\mathbf{R}_j}(\mathbf{R}_j) = \mathbb{P}_{\mathbf{R}_j}(\nabla_{\mathbf{R}_j} \bar{f})$ . The Riemannian Hessian of the function  $f$  at a point  $(\mathbf{R}, \mathbf{t})$  is a linear map,  $\text{Hess } f : \mathcal{T}_{(\mathbf{R}, \mathbf{t})} \mathcal{M} \rightarrow \mathcal{T}_{(\mathbf{R}, \mathbf{t})} \mathcal{M}$  and is defined as shown in Equation (9).

$$\text{Hess } f(\mathbf{R}, \mathbf{t})[\eta_{\mathbf{R}}, \eta_t] = (\mathbb{P}_{\mathbf{R}}(\mathbf{D}\xi_{\mathbf{R}}(\mathbf{R})[\eta_{\mathbf{R}}]), \mathbb{P}_{\mathbf{t}}(\mathbf{D}\xi_{\mathbf{t}}(\mathbf{t})[\eta_t])). \quad (9)$$

Here, the first component  $\mathbb{P}_{\mathbf{R}}(\mathbf{D}\xi_{\mathbf{R}}(\mathbf{R})[\eta_{\mathbf{R}}])$  is equal to

$$(\mathbb{P}_{\mathbf{R}_1}(\mathbf{D}\xi_{\mathbf{R}_1}(\mathbf{R}_1)[\eta_{\mathbf{R}_1}]), \dots, \mathbb{P}_{\mathbf{R}_{d-1}}(\mathbf{D}\xi_{\mathbf{R}_{d-1}}(\mathbf{R}_{d-1})[\eta_{\mathbf{R}_{d-1}}])),$$

where  $\eta_{\mathbf{R}_j} = \mathbf{R}_j \Omega_j$ . The term

$$\mathbf{D}\xi_{\mathbf{x}}(\mathbf{x})[\eta_{\mathbf{x}}] = \lim_{t \rightarrow 0} \frac{\xi(\mathbf{x} + t\eta_{\mathbf{x}}) - \xi(\mathbf{x})}{t}$$

is the classical derivative of the vector field  $\xi(\mathbf{x})$  in the direction  $\eta_{\mathbf{x}}$ .

**The Riemannian trust region method.** Our goal is to minimize the function  $f(\mathbf{R}, \mathbf{t})$  over the product manifold  $\mathcal{M}$ . There exists a generalization of the popular optimization methods on the Riemannian manifolds. Since our problem is locally convex in each variable  $\mathbf{R}_j$ , which we prove in Theorem 6, we employ the Riemannian trust region approach [51]. It requires the Riemannian gradient and the Riemannian Hessian operator for the function  $f$ , which we find as follows. Let  $\bar{f}$  be a function from the set  $\mathbb{V} \times \mathbb{R}^d$  to  $\mathbb{R}$  and defined as  $\bar{f}(\mathbf{R}, \mathbf{t}) = \|\mathbf{TET}^\top(\mathbf{X} - \mathbf{te}^\top) + \mathbf{te}^\top - \mathbf{XP}\|_{\mathbb{F}}^2$ . Its classical gradients with respect to both the variables are given in the Equations (10) and (11). The detailed derivation is given in the Appendices §A1 and §A2.

$$\nabla_{\mathbf{t}} \bar{f} = 2(\mathbf{I}_d - \mathbf{TET}^\top)(2\mathbf{e}^\top \mathbf{e} \mathbf{t} - \mathbf{X} \mathbf{e} - \mathbf{XP} \mathbf{e}). \quad (10)$$

$$\nabla_{\mathbf{R}_j} \bar{f} = -2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top. \quad (11)$$

Here,

$$\mathbf{A} = (\mathbf{XP} - \mathbf{te}^\top)(\mathbf{X} - \mathbf{te}^\top)^\top + (\mathbf{X} - \mathbf{te}^\top)(\mathbf{XP} - \mathbf{te}^\top)^\top$$

which satisfies  $\mathbf{A}^\top = \mathbf{A}$ . Now let the function  $f = \bar{f}|_{\mathcal{M}}$  be the restriction of the function  $\bar{f}$  on the set  $\mathcal{M}$ . We obtain the Riemannian gradient of the function  $f$  at a point  $(\mathbf{R}, \mathbf{t})$  by projecting the Riemannian gradient of the function  $\bar{f}$  over the tangent space  $\mathcal{T}_{(\mathbf{R}, \mathbf{t})}$  at the point  $(\mathbf{R}, \mathbf{t})$ . Since the manifold  $\mathbb{V} \times \mathbb{R}^d$  is an Euclidean space, the Riemannian gradient of the function  $\bar{f}$  is equal to its classical gradient. Therefore, we apply the definition given in Equation (8) in order to find the Riemannian gradient  $\text{grad } f(\mathbf{R}, \mathbf{t})$  of the function  $f$  which we denote as  $(\xi_{\mathbf{R}_1}(\mathbf{R}_1), \dots, \xi_{\mathbf{R}_{d-1}}(\mathbf{R}_{d-1}), \xi_{\mathbf{t}})$  and define in Equations (12) and (13). The detailed derivation is given in the Appendices §A3 and §A4.

$$\xi_{\mathbf{t}}(\mathbf{t}) = 2(\mathbf{I}_d - \mathbf{TET}^\top)(2\mathbf{e}^\top \mathbf{e} \mathbf{t} - \mathbf{X} \mathbf{e} - \mathbf{XP} \mathbf{e}), \quad (12)$$

$$\begin{aligned} \xi_{\mathbf{R}_j}(\mathbf{R}_j) &= -\mathbf{R}_j \left( \prod_{u=1}^j \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &+ \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^j \mathbf{R}_u \right). \end{aligned} \quad (13)$$

We determine the Riemannian Hessian of the function  $f$  using the definition given in Equation (9). In order to determine the  $j^{\text{th}}$  component  $\text{Hess}_{\mathbf{R}_j}(f(\mathbf{R}, \mathbf{t}))[\mathbf{R}_j \Omega_j]$  of the Riemannian Hessian, which is equal to  $\mathbb{P}_{\mathbf{R}_j}(\mathbf{D}\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j \Omega_j])$ , we first find the classical derivative  $\mathbf{D}\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j \Omega_j]$  of the Riemannian gradient  $\xi_{\mathbf{R}_j}(\mathbf{R}_j)$  in the direction  $\mathbf{R}_j \Omega_j$  and then apply the projection operator  $\mathbb{P}_{\mathbf{R}_j}$ . Therefore, the  $j^{\text{th}}$  component  $\text{Hess}_{\mathbf{R}_j}(f(\mathbf{R}, \mathbf{t}))[\mathbf{R}_j \Omega_j]$  of the Riemannian Hessian is equal to

$$\frac{1}{2} \mathbf{R}_j ([\mathbf{B}_1, [\mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j, \Omega_j]] + [[\Omega_j, \mathbf{B}_1], \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j]). \quad (14)$$

The detailed derivation is given in the Appendix §A5. Here  $[\cdot, \cdot]$  is the Lie bracket and defined as  $[\mathbf{U}, \mathbf{V}] = \mathbf{UV} - \mathbf{VU}$  for any two matrices  $\mathbf{U}$  and  $\mathbf{V}$ ,

$$\mathbf{B}_1 = \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top,$$

and

$$\mathbf{B}_2 = \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right).$$

In a similar way, we determine the component,  $\mathbb{P}_{\mathbf{t}}(\mathbf{D}\xi_{\mathbf{t}}(\mathbf{t})[\eta_t])$ , of the Riemannian Hessian which is shown in Equation (15).

$$\text{Hess}_{\mathbf{t}}(f(\mathbf{R}, \mathbf{t}))[\eta_t] = 4n(\mathbf{I}_d - \mathbf{TET}^\top)\eta_t. \quad (15)$$

The detailed derivation is given in the Appendix §A6. Now, we apply the Riemannian-trust-region method using the Riemannian gradient and Hessian defined in Equations (12), (13), (14), and (15) in order to obtain the optimal solution. We use the *manopt* toolbox in order to implement the optimization problem given in Equation (5) for a fixed  $\mathbf{P}$  [52].

### Determining the reflection symmetry hyperplane $\pi$ .

In order to determine the reflection hyperplane  $\pi$ , we use Theorem 2 which states that the normal vector of  $\pi$  lies in the null space of the matrix  $\mathbf{I}_d + (\prod_{u=1}^{d-1} \mathbf{R}_u) \mathbf{E} (\prod_{u=1}^{d-1} \mathbf{R}_u)^\top$  and the optimal translation  $\mathbf{t}$  lies on the hyperplane.

### C. Optimizing Correspondences $\mathbf{P}$

After obtaining the current estimate of the reflection transformation  $(\mathbf{R}, \mathbf{t})$ , we improve the correspondences matrix  $\mathbf{P}$  by solving the problem given in Equation (5) while fixing  $(\mathbf{R}, \mathbf{t})$ . We show that this sub-problem is equivalent to a linear assignment problem, where an *assignment* is a pair  $(i, i')$  of reflection symmetry points.

**Claim 1:** *The optimization problem given in Equation (5) is a linear assignment problem in  $\mathbf{P}$ , for a fixed  $(\mathbf{R}, \mathbf{t})$ .*

**Proof:** Let us consider the cost function in Equation (5) and let  $\mathbf{X}_m = \mathbf{TET}^\top(\mathbf{X} - \mathbf{te}^\top) + \mathbf{te}^\top$ . We have

$$\begin{aligned} \|\mathbf{X}_m - \mathbf{XP}\|_F^2 &= \text{trace}((\mathbf{X}_m - \mathbf{XP})^\top(\mathbf{X}_m - \mathbf{XP})) \\ &= \text{trace}(\mathbf{X}_m^\top \mathbf{X}_m - 2\mathbf{X}_m^\top \mathbf{XP} + \mathbf{X}^\top \mathbf{XPP}^\top). \end{aligned}$$

Since, the first and the third terms (using the fact that the permutation matrices are orthogonal) are not the functions of  $\mathbf{P}$ , the problem of finding the point of minimum of the function  $\|\mathbf{X}_m - \mathbf{XP}\|_F^2$  is identical to the problem of finding the point of maximum of the function  $\text{trace}(\mathbf{X}_m^\top \mathbf{XP})$ . Using the identity  $\text{trace}(\mathbf{A}^\top \mathbf{B}) = \text{vec}(\mathbf{A})^\top \text{vec}(\mathbf{B})$ , we have that  $\text{trace}(\mathbf{X}_m^\top \mathbf{XP}) = \text{vec}(\mathbf{X}^\top \mathbf{X}_m)^\top \text{vec}(\mathbf{P})$ , where the operator  $\text{vec}$  vectorizes a matrix by stacking all the columns successively in a column vector.

Therefore, for a fixed reflection transformation, the problem defined in Equation (5) is equivalent to the problem defined in Equation (16).

$$\begin{aligned} \max_{\mathbf{P} \in \{0,1\}^{n \times n}} \quad & \text{trace}(\mathbf{X}_m^\top \mathbf{XP}) = \text{vec}(\mathbf{X}_m^\top \mathbf{X})^\top \text{vec}(\mathbf{P}) \\ \text{subject to} \quad & \mathbf{Pe} \leq \mathbf{e}, \quad \mathbf{P}^\top \mathbf{e} \leq \mathbf{e}, \end{aligned} \quad (16)$$

which is a standard linear assignment problem.  $\square$

**Claim 2:** *The problem defined in Equation (16) is an integer linear program.*

**Proof:** Let  $\mathbf{v}_1$  be a vector of size  $n^2 \times 1$  with the first  $n$  coordinates equal to one and the last  $n(n-1)$  coordinates equal to zero. Let  $\mathbf{e}_1$  be a vector of size  $n \times 1$  with all the coordinates equal to zero except the first coordinate which is equal to one. Let  $\mathbf{v}_2 = [\mathbf{e}_1^\top \quad \mathbf{e}_1^\top \quad \dots \quad \mathbf{e}_1^\top]^\top$  be a vector of size  $n^2 \times 1$ . Now let us construct the matrices  $\mathbf{A}_1$  and  $\mathbf{A}_2$ , each of size  $n \times n^2$ , such that the  $i^{\text{th}}$  row of the matrix  $\mathbf{A}_1$  is equal to the row vector  $\text{cs}(\mathbf{v}_1^\top, n(i-1))$  and the  $i^{\text{th}}$  row of the matrix  $\mathbf{A}_2$  is equal to the row vector  $\text{cs}(\mathbf{v}_2^\top, i-1)$ . Here  $\text{cs}(\mathbf{v}^\top, i)$  is a row vector obtained by circularly shifting any row vector  $\mathbf{v}^\top$  right by  $i$  coordinates.

Now, it is trivial to verify that the constraint  $\mathbf{P}^\top \mathbf{e} \leq \mathbf{e}$  is equivalent to  $\mathbf{A}_1 \text{vec}(\mathbf{P}) \leq \mathbf{e}$  and the constraint  $\mathbf{Pe} \leq \mathbf{e}$  is equivalent to  $\mathbf{A}_2 \text{vec}(\mathbf{P}) \leq \mathbf{e}$ . Therefore, the problem

defined in Equation (16) is equivalent to the problem defined in Equation (17).

$$\begin{aligned} \max_{\mathbf{a} \in \{0,1\}^{n^2 \times 1}} \quad & \text{vec}(\mathbf{X}_m^\top \mathbf{X})^\top \mathbf{a} \\ \text{subject to} \quad & [\mathbf{A}_1^\top \quad \mathbf{A}_2^\top]^\top \mathbf{a} \leq [\mathbf{e}^\top \quad \mathbf{e}^\top]^\top \end{aligned} \quad (17)$$

which is an integer linear program with  $\mathbf{a} = \text{vec}(\mathbf{P})$ .  $\square$

**Solving the ILP.** Since ILP is an NP-complete problem, there may not exist a polynomial time algorithm to find the optimal solution. We relax this ILP to a linear program by converting the constraint  $\mathbf{a} \in \{0,1\}^{n^2 \times 1}$  into  $\mathbf{a} \in [0,1]^{n^2 \times 1}$ . Now, the above ILP becomes a linear program. We first solve this LP using the Karmarkar's algorithm in [53] which takes  $O(n^{3.5})$  time. The solution  $\mathbf{a}^* = [a_1^* \quad a_2^* \quad \dots \quad a_{n^2}^*]^\top$  of this LP belongs to  $[0,1]^{n^2 \times 1}$  which is a continuous solution. However, our final solution  $\mathbf{a}^f = [a_1^f \quad a_2^f \quad \dots \quad a_{n^2}^f]^\top$  of the proposed ILP should be a discrete solution. We follow the rounding approach, as explained in ([54], ch. 11). The  $i$ -th element  $a_i^f$  of the final solution is equal to 1, if  $a_i^* \geq 0.5$  and equal to 0, if  $a_i^* < 0.5$ . This solution  $\mathbf{a}^f$  may not be the optimal solution because according to [54],  $\text{vec}(\mathbf{X}_m^\top \mathbf{X})^\top \mathbf{a}^f \geq \frac{1}{2} \times \text{vec}(\mathbf{X}_m^\top \mathbf{X})^\top \mathbf{a}^{OPT}$ . Here,  $\mathbf{a}^{OPT}$  is the optimal solution of the above ILP.

### D. Convergence Analysis

We derive the essential results in order to prove that the alternating optimization framework converges.

**Theorem 3:** *The cost function  $f(\mathbf{R}, \mathbf{t}, \mathbf{P})$  is convex in the variable  $\mathbf{t}$ .*

**Proof:** In order to prove this, we prove that the Riemannian Hessian of the function  $f$  with respect to the variable  $\mathbf{t}$  is a positive semi-definite (PSD) matrix. Let  $\boldsymbol{\eta}_t = [\eta_1 \quad \eta_2 \quad \dots \quad \eta_d]^\top \in \mathbb{R}^d$ . Then using the definition of Riemannian metric, we have

$$\langle \boldsymbol{\eta}_t, \text{Hess}_t(f)[\boldsymbol{\eta}_t] \rangle_t = \boldsymbol{\eta}_t^\top \text{Hess}_t(f)[\boldsymbol{\eta}_t].$$

Now, using the Riemannian Hessian  $\text{Hess}_t(f)[\boldsymbol{\eta}_t]$  defined in Equation (15), we have that

$$\boldsymbol{\eta}_t^\top \text{Hess}_t(f)[\boldsymbol{\eta}_t] = \boldsymbol{\eta}_t^\top \boldsymbol{\eta}_t - (\mathbf{T}^\top \boldsymbol{\eta}_t)^\top \mathbf{E}(\mathbf{T}^\top \boldsymbol{\eta}_t)$$

Now let  $\mathbf{q} = \mathbf{T}^\top \boldsymbol{\eta}_t$ . Then, we obtain

$$\begin{aligned} \boldsymbol{\eta}_t^\top \text{Hess}_t(f)[\boldsymbol{\eta}_t] &= \boldsymbol{\eta}_t^\top \boldsymbol{\eta}_t - \mathbf{q}^\top \mathbf{E} \mathbf{q} \\ &= \|\boldsymbol{\eta}_t\|_2^2 - \sum_{u=1}^{d-1} q_u + q_d^2 = \|\boldsymbol{\eta}_t\|_2^2 - \|\mathbf{q}\|_2^2 + 2q_d^2. \end{aligned}$$

Now, we know that  $\mathbf{T}\mathbf{T}^\top = \mathbf{I}$ . Hence, we have

$$\|\mathbf{q}\|_2^2 = \mathbf{q}^\top \mathbf{q} = \boldsymbol{\eta}_t^\top \mathbf{T}\mathbf{T}^\top \boldsymbol{\eta}_t = \boldsymbol{\eta}_t^\top \boldsymbol{\eta}_t = \|\boldsymbol{\eta}_t\|_2^2.$$

Therefore,

$$\|\boldsymbol{\eta}_t\|_2^2 - \|\mathbf{q}\|_2^2 = 0 \Rightarrow \boldsymbol{\eta}_t^\top \text{Hess}_t(f)[\boldsymbol{\eta}_t] = 2q_d^2 \geq 0. \square$$

**Theorem 4:** *At the critical point, the matrix  $\mathbf{T}^* = \prod_{u=1}^d \mathbf{R}_u^*$  contains the eigenvectors of the matrix  $\mathbf{A}$  as columns.*

**Proof:** At the critical point, the Riemannian gradient given in Equation (13) vanishes. Therefore,  $\xi_{\mathbf{R}_j}(\mathbf{R}_j) = \mathbf{0}_{d \times d}$ . Now pre-multiplying it with  $(\prod_{u=1}^j \mathbf{R}_u) \mathbf{R}_j^\top$  and then post-multiplying with  $(\prod_{u=j+1}^{d-1} \mathbf{R}_u)$ , we achieve

$$\mathbf{A} \mathbf{T}^* \mathbf{E} = \mathbf{T}^* \mathbf{E} (\mathbf{T}^*)^\top \mathbf{A} \mathbf{T}^* \Rightarrow (\mathbf{T}^*)^\top \mathbf{A} \mathbf{T}^* \mathbf{E} = \mathbf{E} (\mathbf{T}^*)^\top \mathbf{A} \mathbf{T}^*.$$

Now, let  $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{q}_2 \\ \mathbf{q}_3 & q_4 \end{bmatrix} = (\mathbf{T}^*)^\top \mathbf{A} \mathbf{T}^*$  be a matrix. Then, we have  $\mathbf{Q} \mathbf{E} = \mathbf{E} \mathbf{Q}$ . Therefore,

$$\begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{q}_2 \\ \mathbf{q}_3 & q_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{d-1} & \mathbf{0}_{d-1} \\ \mathbf{0}_{d-1}^\top & -1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{d-1} & \mathbf{0}_{d-1} \\ \mathbf{0}_{d-1}^\top & -1 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^\top & \mathbf{q}_2 \\ \mathbf{q}_3 & q_4 \end{bmatrix} \\ \Rightarrow \mathbf{q}_2 = \mathbf{0}_{d-1}, \mathbf{q}_3 = \mathbf{0}_{d-1}, \mathbf{Q}_1 \mathbf{I}_{d-1} = \mathbf{I}_{d-1} \mathbf{Q}_1.$$

Since,  $\mathbf{I}_{d-1}$  is a diagonal matrix and the equality  $\mathbf{Q}_1 \mathbf{I}_{d-1} = \mathbf{I}_{d-1} \mathbf{Q}_1$  holds true, it is easy to prove that  $\mathbf{Q}_1$  is a diagonal matrix. Therefore, the matrix  $\mathbf{Q}$  is also diagonal. The *spectral theorem* states that every real symmetric matrix has eigenvalue decomposition with real eigenvalues and orthogonal eigenvectors. Here, we have observed that the matrix  $\mathbf{A}$  is a real symmetric matrix and satisfies  $\mathbf{Q} = (\mathbf{T}^*)^\top \mathbf{A} \mathbf{T}^*$ , where the matrix  $\mathbf{Q}$  is a diagonal matrix and the matrix  $\mathbf{T}^*$  is an orthogonal matrix. Therefore, the matrix  $\mathbf{T}^*$  is the matrix containing the eigenvectors of the matrix  $\mathbf{A}$ . In Theorem 5, we prove that the order of stacking eigenvectors of  $\mathbf{A}$  as columns of  $\mathbf{T}^*$  affects the convexity of the problem.  $\square$

**Theorem 5:** *The cost function  $f(\mathbf{R}, \mathbf{t}, \mathbf{P})$  is locally convex in each rotation matrix  $\mathbf{R}_j$ .*

**Proof:** In order to show the local convexity in  $\mathbf{R}_j$ , we have to show that the value  $\langle \mathbf{R}_j \Omega_j, \mathbf{H}[\mathbf{R}_j \Omega_j] \rangle_{\mathbf{R}_j} \geq 0$  in the neighborhood of the optimal angle  $\theta_j^*$ . Here,  $\mathbf{H}[\mathbf{R}_j \Omega_j] = \text{Hess}_{\mathbf{R}_j}(f(\mathbf{R}, \mathbf{t}))[\mathbf{R}_j \Omega_j]$ . By using the Riemannian metric defined in Equation (7), we have

$$\langle \mathbf{R}_j \Omega_j, \mathbf{H}[\mathbf{R}_j \Omega_j] \rangle_{\mathbf{R}_j} = \text{trace}(\Omega_j^\top \mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j \Omega_j]).$$

By using Equation (14), the matrix  $\mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j \Omega_j]$  is equal to

$$0.5[\mathbf{B}_1, [\mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j, \Omega_j]] + 0.5[[\Omega_j, \mathbf{B}_1], \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j].$$

In the Appendix §A7, we show that the  $\text{trace}(\Omega_j^\top \mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j \Omega_j])$  is equal to

$$4 \times \text{trace}(\mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j (\Omega_j \mathbf{B}_1 \Omega_j - \Omega_j \Omega_j \mathbf{B}_1)). \quad (18)$$

We visualize this term for  $d = 2$ . For  $d = 2$ , the matrix  $\Omega = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}$ ,  $\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ , and let  $\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_2 & a_3 \end{bmatrix}$  and  $\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ . We have that

$$\langle \mathbf{R} \Omega, \mathbf{H}[\mathbf{R} \Omega] \rangle_{\mathbf{R}} = 8a_2 \theta^2 \sin(2\theta) + 4\theta^2 \cos(2\theta)(a_1 - a_3).$$

In Fig. 3, we plot the value  $\frac{\langle \mathbf{R} \Omega, \mathbf{H}[\mathbf{R} \Omega] \rangle_{\mathbf{R}}}{\theta^2}$  against the initialization angle  $\theta$  for six reflection symmetry patterns having different orientations for symmetry axis. We observe that the PSD values are positive in the proximity of the optimal angles. Therefore, it is locally convex. We further observe that this quantity is maximum at the optimal angle. We also observe that, if  $\theta$  is the symmetry axis orientation, then the PSD value becomes positive in the proximity of  $\theta$  and  $\theta + 180^\circ$ . The

reason for the second range is that, if  $\theta$  is the slope of a line, then  $\theta + 180^\circ$  is also the slope of the same line.

In Theorem 4, we claimed that the order in which the eigenvectors are stacked as columns of the matrix  $\mathbf{R}$  affects the local convexity. We prove it as follows. At the critical point, we have that  $\mathbf{R}^\top \mathbf{A} \mathbf{R} = \text{diag}(d_1, d_2)$ . We note that  $\Omega_j \mathbf{B}_1 \Omega_j - \Omega_j \Omega_j \mathbf{B}_1 = \mathbf{E}$  for  $d = 2$ . Now from Equation (18), we achieve

$$\langle \mathbf{R} \Omega, \mathbf{H}[\mathbf{R} \Omega] \rangle_{\mathbf{R}} = d_1 - d_2 \Rightarrow d_1 \geq d_2.$$

Therefore, the first column of the matrix  $\mathbf{R}^*$  should be the eigenvector corresponding to the maximum eigenvalue and the second column of the matrix  $\mathbf{R}^*$  should be the eigenvector corresponding to the minimum eigenvalue of the matrix  $\mathbf{A}$ .  $\square$

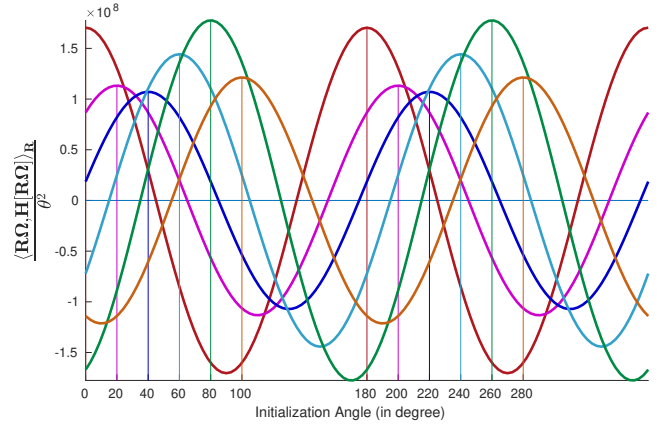


Fig. 3. Illustration of the local convexity. The value  $\frac{\langle \mathbf{R} \Omega, \mathbf{H}[\mathbf{R} \Omega] \rangle_{\mathbf{R}}}{\theta^2}$  against the initialization angle  $\theta$  for 6 reflection symmetry patterns having different orientations,  $\{0^\circ, 20^\circ, 40^\circ, 60^\circ, 80^\circ, 100^\circ\}$  for symmetry axis. The PSD value (divided by  $\theta^2$ ) is positive in the proximity of the optimal angle.

**Theorem 6:** *The proposed alternating framework converges to the global minimum if the initialization of the rotation matrices  $\mathbf{R}_1, \dots, \mathbf{R}_{d-1}$  are within the proximity of the optimal rotation matrices and initialization of the translation  $\mathbf{t}$  is any random vector.*

**Proof:** We observe that the proposed alternation framework is basically the block coordinate descent (BCD) method, where  $(\mathbf{R}_1, \dots, \mathbf{R}_{d-1}, \mathbf{t})$  and  $\mathbf{P}$  are two blocks of coordinates. According to [55], the BCD method converges if the cost function is convex in each block of coordinates. Here, we have seen that the cost function is convex in the coordinates  $\mathbf{t}$  (Theorem 3), convex in the coordinates  $\mathbf{P}$  on the relaxed domain  $[0, 1]^{n \times n}$ , and locally convex in the coordinates  $(\mathbf{R}_1, \dots, \mathbf{R}_{d-1})$  (Theorem 5). This implies that if the initialization of  $(\mathbf{R}_1, \dots, \mathbf{R}_{d-1})$  is within the proximity of the optimal solution, then the alternating framework converges to the global minimum. We experimentally show this theorem for the case  $d = 2$ . We use the dataset for  $d = 2$  with  $\sigma = 0$  as mentioned in §V-B. In Fig. 4, we plot the error (averaged over all optimal angles) at the convergence point against the initialization angles for the case  $d = 2$  (we shift the error vectors for different optimal angles so that the optimal angle is always  $90^\circ$ ). We observe that the variance becomes zero for initialization angle  $\theta_0 \in (90^\circ - 12^\circ, 90^\circ + 9^\circ)$  and  $\theta_0 \in (270^\circ - 12^\circ, 270^\circ + 9^\circ)$ . The reason for the second range

is that, if  $\theta$  is the slope of a line, then  $\theta + 180^\circ$  is also the slope of the same line.  $\square$

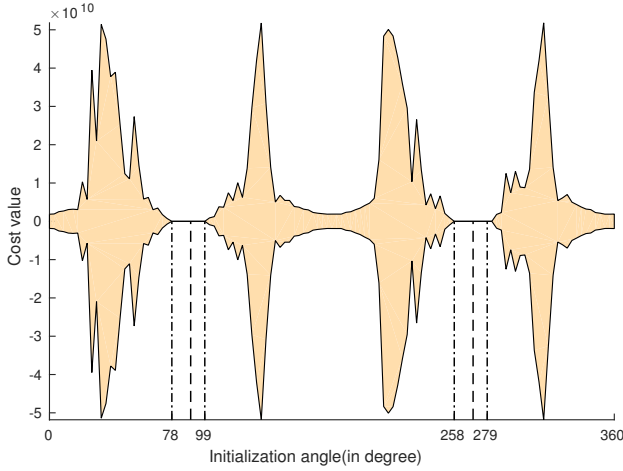


Fig. 4. We plot the error at the convergence point against the initialization angles for the case  $d = 2$  (we shift the error vectors for different optimal angles so that the optimal angle is always  $90^\circ$ ). We observe that the variance in the error becomes zeros for initialization angle  $\theta_0 \in (90^\circ - 12^\circ, 90^\circ + 9^\circ)$  and  $\theta_0 \in (270^\circ - 12^\circ, 270^\circ + 9^\circ)$ .

In summary, in order to obtain the optimal  $(\mathbf{R}^*, \mathbf{t}^*, \mathbf{P}^*)$ , we follow Algorithm 1.

---

#### Algorithm 1

---

- 1: Input: Set of points  $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$ .
  - 2: Initialize angles  $\theta_0$  and translation  $\mathbf{t}$ .
  - 3: Solve the ILP defined in Equation (17) for  $\mathbf{P}$ .
  - 4: For this  $\mathbf{P}$ , solve for  $(\mathbf{R}, \mathbf{t})$  using the Riemannian-trust-region method using the Riemannian gradient and Hessian defined in Equations (12), (13), (14), and (15).
  - 5: Keep iterating steps 3 and 4 till convergence.
  - 6: Output: The optimal  $\mathbf{R}_1^*, \mathbf{R}_2^*, \dots, \mathbf{R}_{d-1}^*$  and  $\mathbf{t}^*$ .
- 

**Initialization Strategy:** In the Theorem 5, we have shown that  $f(\mathbf{R}, \mathbf{t}, \mathbf{P})$  is locally convex in rotation matrix  $\mathbf{R}$ . Therefore, Algorithm 1 converges to the global minimum if we initialize the rotation matrix in the proximity of the global solution. Hence, we approximate the initial  $\mathbf{R}$  by finding a small set of candidate pairs of mirror symmetric points. We discuss the proposed approach for finding a small set of candidate pairs of mirror symmetric points as follows.

Let us consider the input set  $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^n$ . We propose a randomized approach to find a small set of candidate pairs of mirror symmetric points. We select two points,  $\mathbf{x}_p$  and  $\mathbf{x}_q$ , uniformly at random from the set  $\mathcal{S}$ . Let  $\mathbf{x}_{p'}$  and  $\mathbf{x}_{q'}$  be their actual mirror images, respectively. We then construct two sets,  $\mathcal{P} = \{(\mathbf{x}_p, \mathbf{x}_i)\}_{i=1, i \neq p, q}^n$  and  $\mathcal{Q} = \{(\mathbf{x}_q, \mathbf{x}_i)\}_{i=1, i \neq q, p}^n$  of pairs of points. Given the sets  $\mathcal{P}$  and  $\mathcal{Q}$ , our goal is to find the pairs  $(\mathbf{x}_p, \mathbf{x}_{p'})$  and  $(\mathbf{x}_q, \mathbf{x}_{q'})$ . It is trivial to observe that  $(\mathbf{x}_p, \mathbf{x}_{p'}) \in \mathcal{P}$  and  $(\mathbf{x}_q, \mathbf{x}_{q'}) \in \mathcal{Q}$ . We note that each pair of points define its own symmetry plane, the one which is perpendicular to the line segment joining the two points and passing through the mid-point of this line segment. Now, if the pairs  $(\mathbf{x}_p, \mathbf{x}_{p'})$  and  $(\mathbf{x}_q, \mathbf{x}_{q'})$  are true pairs then both the

reflection planes, defined by these two pairs, should be the same. For each pair  $(\mathbf{x}_p, \mathbf{x}_i) \in \mathcal{P}$ , we keep sampling a pair  $(\mathbf{x}_q, \mathbf{x}_j) \in \mathcal{Q}$  uniformly at random without replacement until the reflection planes defined by these two pairs are the same. We determine whether the two reflection planes, defined by these two pairs,  $\pi_{pi} : \boldsymbol{\eta}_{pi}^\top \mathbf{x} - c_{pi} = 0$  and  $\pi_{qj} : \boldsymbol{\eta}_{qj}^\top \mathbf{x} - c_{qj} = 0$  are the same if the conditions,  $\cos^{-1}(\boldsymbol{\eta}_{pi}^\top \boldsymbol{\eta}_{qj}) \leq \epsilon_\theta$  and  $\frac{\min\{d_q, d_j\}}{\max\{d_q, d_j\}} \geq 1 - \epsilon_d$  are true. Here,  $\boldsymbol{\eta}_{pi} = \frac{\mathbf{x}_p - \mathbf{x}_i}{\|\mathbf{x}_p - \mathbf{x}_i\|_2}$  is the normal vector to the plane  $\pi_{pi}$ ,  $c_{pi} = \boldsymbol{\eta}_{pi}^\top (\frac{\mathbf{x}_p + \mathbf{x}_i}{2})$  is the distance of the origin from the plane  $\pi_{pi}$ ,  $\boldsymbol{\eta}_{qj} = \frac{\mathbf{x}_q - \mathbf{x}_j}{\|\mathbf{x}_q - \mathbf{x}_j\|_2}$  is the normal vector to the plane  $\pi_{qj}$ ,  $c_{qj} = \boldsymbol{\eta}_{qj}^\top (\frac{\mathbf{x}_q + \mathbf{x}_j}{2})$  is the distance of the origin from the plane  $\pi_{qj}$ ,  $d_q = |\boldsymbol{\eta}_{pi}^\top \mathbf{x}_q - c_{pi}|$ , and  $d_j = |\boldsymbol{\eta}_{pi}^\top \mathbf{x}_j - c_{pi}|$ .

We repeat the above procedure ten times. With this, we get a set of 20 (2 for each run) candidate pairs of mirror symmetric points. Since we consider the case where only a single symmetric object is present in the input set, we consider the median plane of the 20 planes defined by the above computed 20 candidate pairs. Now, we use the normal  $\boldsymbol{\eta}$  to this median plane for initialization. We also initialize the initial translation vector  $\mathbf{t}$  as the median of the mid-points of the line segment joining the points of the candidate pairs of the mirror symmetric points.

First, we subtract each data point of the point cloud from the estimated  $\mathbf{t}$  of the point cloud. This ensures that the reflection symmetry plane passes through the origin. Now, we know the unit normal to the reflection symmetry plane. Therefore, we use the Householder transform to reflect each point which is  $\mathbf{x}_{i'} = (\mathbf{I} - 2\boldsymbol{\eta}\boldsymbol{\eta}^\top)\mathbf{x}_i$ . Therefore, we have the matrix  $\mathbf{X}$  containing the original point cloud and the matrix  $\mathbf{X}_m$  containing the reflected point cloud about the estimated reflection symmetry plane. Now, using  $\mathbf{X}$  and  $\mathbf{X}_m$ , we solve the linear assignment problem, defined in Equation (17) to find the matrix  $\mathbf{P}$ . Now, we use these approximate correspondences to estimate the reflection symmetry plane as step 4 of Algorithm 1.

#### IV. TIME COMPLEXITY

There are two main steps involved in our algorithm. The first one is to solve for reflection symmetry transformation matrices  $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{d-1}, \mathbf{t}$  using the Riemannian trust region [51]. The second step is to find the pairs of reflective symmetric points using an integer linear program. The time complexity of Riemannian trust region method is  $O(nd^2)$ . Since solving integer linear program is an  $NP$ -complete problem, we first relax it to a linear program (as discussed at the end of §III-C). The time complexity of solving a linear program is polynomial in the number of points in the point cloud. We use the Karmarkar's algorithm in [53] which has the time complexity of  $O(n^{3.5})$ . Therefore, the overall complexity of our approach is polynomial in the number of points in the point cloud which is equal to  $O(nd^2) + O(n^{3.5}) \approx O(n^{3.5})$ , since  $d \ll n$ . It takes around 38.4 seconds ( $d = 3$ ) to find the symmetry plane and all the pairs of mirror symmetric points in a point cloud with 500 points using MATLAB on a Linux machine with i7-7500U CPU @ 2.70GHz, and 16GB RAM.



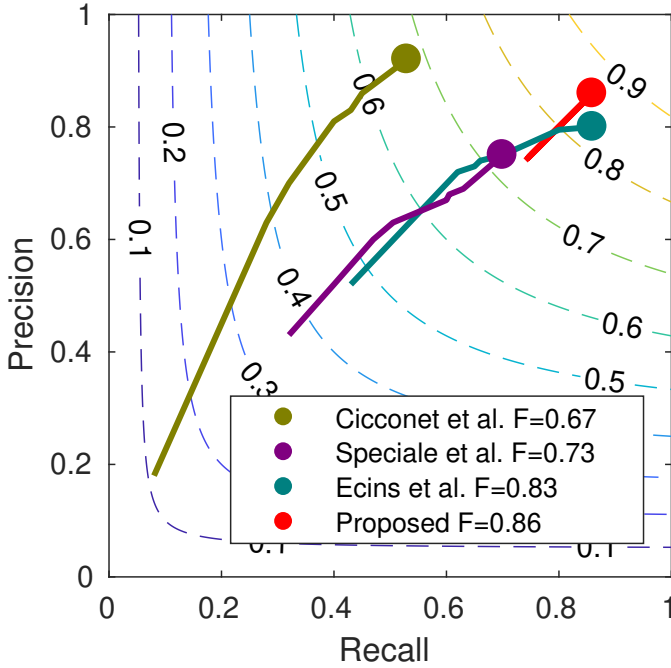


Fig. 5. Recall vs Precision curves for methods Cicconet *et al.* [13], Ecins *et al.* [12], Speciale *et al.* [11], and the proposed approach on the dataset given [56]. We show the maximum F-score for each method in the legends and corresponding points on the precision vs. recall curve using the same colored point.

## V. EVALUATION AND RESULTS

### A. Evaluation of Reflection Symmetry Plane

In order to evaluate the performance of reflection symmetry plane detection, we compare the performance of our approach with the performance of the methods in [12], [13], and [11]. We compare the detected plane of reflection symmetry to that of these methods on the dataset in [56] with F-score as the evaluation metric proposed in [56]. The dataset given in [56] contains models of 1354 3D real world objects in which the ground-truth plane of reflection symmetry is provided for all the objects.

Speciale *et al.* proposed a Hough transform voting based approach [11]. Ecins *et al.* proposed an ICP based approach [12]. First, they initialize the reflection symmetry plane and then iteratively update the reflection symmetry plane using the Levenberg-Marquardt solver till convergence. They have further used the normals at each point to reject outliers points. Therefore, they need oriented point clouds, i.e., normal at each point be given. Cicconet *et al.* first reflected the original point cloud about an arbitrary reflection plane and then used the ICP algorithm to align the original point cloud and the reflected point cloud [13]. Then, they determine the reflection symmetry plane.

In order to evaluate the accuracy of detecting reflection symmetry plane for each method, we find the precision and recall rates and the  $F$ -score. According to [56], the precision and the recall rates are defined as  $P = \frac{TP}{TP+FP}$ ,  $R = \frac{TP}{TP+FN}$ , respectively. The  $F$ -Score is defined as  $F = \frac{2RP}{R+P}$ . According to [56],  $TP$  is equal to the number of correctly estimated reflection symmetry planes,  $FP$  is equal to the number of

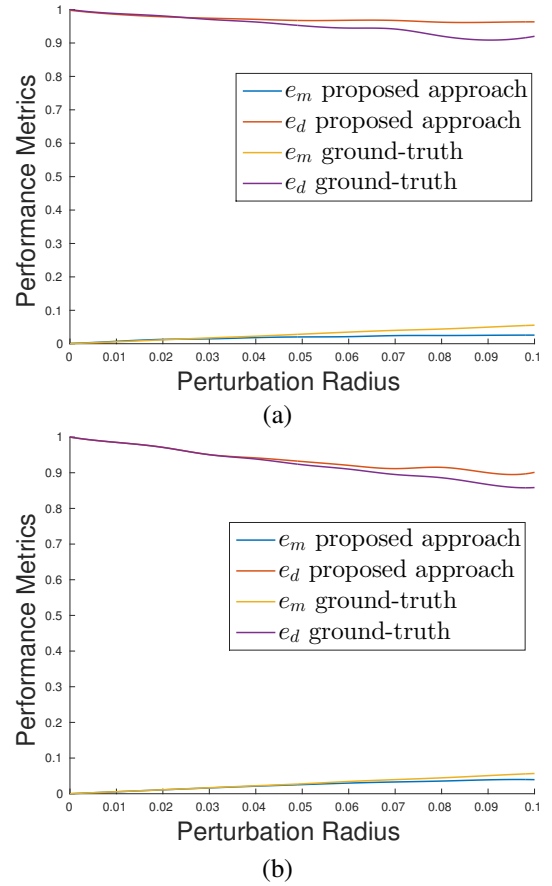


Fig. 6. The values  $e_d$  and  $e_m$  vs the perturbation radius  $\sigma^2$ . (a)  $d=2$ , and (b)  $d=3$ . We observe that the performance measure quantities  $e_d$  and  $e_m$  remain close to that of the ground truth quantities.

incorrectly estimated reflection symmetry planes, and  $FN$  is equal to the number of ground-truth reflection symmetry planes which are not detected. According to [56], a detected plane of reflective symmetry is declared to be correct or incorrect as follows. Let  $\mathbf{x}_1^e$ ,  $\mathbf{x}_2^e$ , and  $\mathbf{x}_3^e$  be three points on the detected plane of reflection symmetry. Let  $\mathbf{x}_1^g$ ,  $\mathbf{x}_2^g$ , and  $\mathbf{x}_3^g$  be three points on the ground truth plane of reflection symmetry of the underlying symmetric object. These three points on the plane of reflection symmetry planes are any three points from the four points of intersection of the plane of reflection symmetry with the bounding box of the underlying reflective symmetric object. Now, according to [56], the detected plane of reflection symmetry is declared correct if the angle between the normal of the detected plane of reflection symmetry, which is defined as  $\boldsymbol{\eta}_e = (\mathbf{x}_1^e - \mathbf{x}_2^e) \times (\mathbf{x}_1^e - \mathbf{x}_3^e)$ , and the normal of the ground truth plane of reflection symmetry, which is defined as  $\boldsymbol{\eta}_g = (\mathbf{x}_1^g - \mathbf{x}_2^g) \times (\mathbf{x}_1^g - \mathbf{x}_3^g)$ , is less than a predefined threshold, i.e.,  $\cos^{-1} \left( \frac{|\boldsymbol{\eta}_e^\top \boldsymbol{\eta}_g|}{\|\boldsymbol{\eta}_e\|_2 \|\boldsymbol{\eta}_g\|_2} \right) < t_\theta$ . Furthermore, according to [56], the distance between the center of the detected plane of reflection symmetry, which defined as  $\mathbf{c}_e = \frac{\mathbf{x}_1^e + \mathbf{x}_2^e}{2}$ , from the ground truth plane of reflection symmetry is less than a predefined threshold, i.e.,  $\frac{|\mathbf{c}_e^\top \boldsymbol{\eta}_g - \boldsymbol{\eta}_g^\top \mathbf{x}_1^e|}{\|\boldsymbol{\eta}_g\|_2} < t_d$ . In order to find the precision vs. recall curve, we change the threshold for angle as  $t_\theta \in [0, 45^\circ]$  and the threshold for distance as

$t_d \in [0, 2s]$ . Here,  $s = \min\{\|\mathbf{x}_1^e - \mathbf{x}_2^e\|_2, \|\mathbf{x}_1^e - \mathbf{x}_3^e\|_2, \|\mathbf{x}_1^e - \mathbf{x}_2^g\|_2, \|\mathbf{x}_1^e - \mathbf{x}_3^g\|_2\}$ . In Fig. 5, we plot the recall vs. precision curves for the methods in [13], [12], [11], and the proposed approach on the dataset given in [56]. We show the maximum  $F$ -score for each method in the legends. The maximum  $F$ -score for [12] is equal to 0.83, for [13] is equal to 0.67, for [11] is equal to 0.73, and for the proposed approach is equal to 0.86.

### B. Robustness to Perturbations

In order to measure the qualitative performance of the proposed approach, we investigate the following two errors which are functions of the perturbation radius  $\sigma^2$ :

$$e_d = \frac{1}{n} \sum_{i=1}^n |\langle \hat{\mathbf{z}}_i, \hat{\mathbf{v}} \rangle| \quad \text{and} \quad e_m = \frac{1}{n} \sum_{i=1}^n |\hat{\mathbf{v}}^\top \mathbf{x}_i^m + w_0|.$$

The error  $e_d$  represents how well the vectors, along the line segments joining the estimated reflection symmetry points, align with the normal to the hyperplane  $\pi$  at convergence. The error  $e_m$  represents how well the mid-points of line segments joining reflection symmetry points lie on the estimated hyperplane  $\pi$ . Here,  $\hat{\mathbf{z}}_i = \frac{\mathbf{x}_i - \mathbf{x}_{i'}}{\|\mathbf{x}_i - \mathbf{x}_{i'}\|_2}$ ,  $\hat{\mathbf{v}}$  is the unit normal to the hyperplane  $\pi$ ,  $\mathbf{x}_i^m = \frac{\mathbf{x}_i + \mathbf{x}_{i'}}{2}$ , and  $w_0$  is the distance of the hyperplane  $\pi$  from the origin. In Fig. 6, we show the errors  $e_d$  and  $e_m$  against the perturbation radius  $\sigma^2$ . We observe that the values  $e_m$  and  $e_d$  for the proposed approach are close to that of the ground-truth reflection symmetry even as the value of  $\sigma^2$  increases. We construct the following dataset to perform the above experiment. Let  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\frac{n}{2}}\}$  be the randomly chosen  $\frac{n}{2}$  points. Given the reflection transformations  $\{\mathbf{R}_1, \dots, \mathbf{R}_{d-1}, \mathbf{t}\}$ , we reflect these points using the *Definition 1* in order to get the final symmetric set  $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\frac{n}{2}}, \mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_{\frac{n}{2}}\}$ . Then, we perturb each point with random noise as  $\mathbf{x} \leftarrow \mathbf{x} + \mathcal{N}(\mathbf{0}_d, \text{diag}(\sigma^2, \sigma^2, \dots, \sigma^2))$ ,  $\forall \mathbf{x} \in \mathcal{S}$ . Here,  $\sigma^2$  is the perturbation radius and the perturbation is different for each point. For the case  $d = 2$ , we create sets containing reflection symmetry patterns with  $n \in \{50, 100, 150, 200, 250, 300\}$  with  $0 \leq x, y \leq 1$ . For each  $n$ , we take 19 different symmetry axis orientations in the range from  $-90^\circ$  to  $90^\circ$  with step size of  $10^\circ$ . We choose  $\sigma^2 \in \{0, 0.01, 0.02, \dots, 0.1\}$  to get 11 different perturbations. In total, we have 1254 sets for the evaluation. In Fig. 7, we show an example point set from this dataset. For the case  $d = 3$ , we create reflective symmetric sets with  $n \in \{50, 100, 150, 200, 250, 300\}$  with  $0 \leq x, y \leq 1$ . For each  $n$ , we take 16 different symmetry plane orientations by considering  $\theta_1 \in \{-30^\circ, 0^\circ, 35^\circ, 80^\circ\}$  and  $\theta_2 \in \{-30^\circ, 0^\circ, 35^\circ, 80^\circ\}$ . We choose  $\sigma^2 \in \{0, 0.01, \dots, 0.1\}$ . In total, we obtain 1056 point sets.

### C. Evaluation in Higher Dimensional Data

**Datasets.** Since datasets for higher dimensions ( $d > 3$ ) are not available with ground-truth reflection symmetry, we synthetically create datasets as follows. For the case  $d = 6$  and  $d = 8$ , we create mirror symmetric point clouds using *Definition 1*, with  $n \in \{50, 100, 150, 200, 250, 300\}$  and

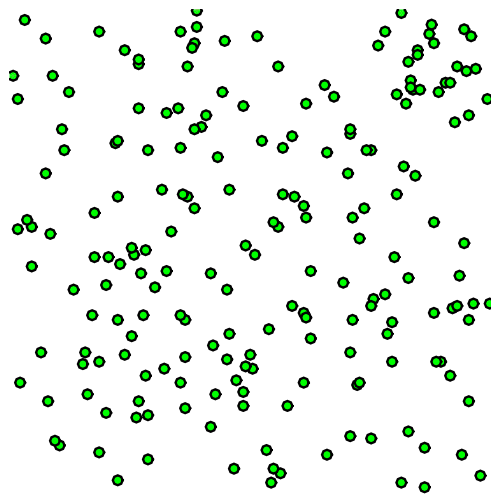


Fig. 7. An example point set for  $d = 2$ .

$0 \leq x, y \leq 1$ . For each  $n$ , we take 20 random symmetry plane normals. We choose  $\sigma^2 \in \{0, 0.02, 0.04, \dots, 0.1\}$  to get 6 different perturbations. In total, we have 720 sets for evaluation. For all these point clouds, we have the ground-truth correspondences between the symmetric points and the normals to the ground-truth symmetry planes.

**Evaluation of correspondences.** In order to evaluate the performance, we measure the correspondence rate which is the number of correct correspondences out of the estimated correspondences. Let  $(i, i'_e)$  be the estimated correspondence and let  $(i, i'_g)$  be the ground-truth correspondence. Then, we decide if the estimated correspondence  $(i, i'_e)$  is correct based on a distance threshold  $\tau_d$ . If the distance  $\|\mathbf{x}_{i'_e} - \mathbf{x}_{i'_g}\|_2$  between the points  $\mathbf{x}_{i'_e}$  and  $\mathbf{x}_{i'_g}$  is less than the distance threshold  $\tau_d$ , then the correspondence  $(i, i'_e)$  is correct and otherwise, incorrect. For a given threshold  $\tau_d$ , we count the correspondences  $(i, i'_e)$  for which the condition  $\|\mathbf{x}_{i'_e} - \mathbf{x}_{i'_g}\|_2 < \tau_d$  holds true. In Fig. 8, we show the correspondence rate vs the distance threshold curves for the different perturbation radius  $\sigma^2 \in \{0, 0.02, 0.04, \dots, 0.1\}$  and for  $d = 6$  and  $d = 8$ . We vary the distance threshold as  $\tau_d \in \{0, 0.01, 0.02, \dots, 0.34\}$ . We observe that the correspondence rate increases as the distance threshold increases and the correspondence rate decreases as the perturbation radius increases for both  $d = 6$  and  $d = 8$ .

**Evaluation of symmetry plane.** To evaluate the performance of the reflection plane detection in higher dimensional point clouds ( $d > 3$ ), instead of finding  $d - 1$  points on the estimated hyperplane (since finding  $d - 1$  points could be difficult), we measure the distance between their normals. Without loss of generality, we prepare the dataset such that the reflection symmetry plane passes through the origin. Now, let  $\boldsymbol{\eta}^g$  and  $\boldsymbol{\eta}^e$  be the unit normals to the ground-truth and the estimated reflection symmetry planes, respectively. Then, we declare the estimated reflection symmetry plane to be correct, if  $\cos^{-1}(|(\boldsymbol{\eta}^g)^\top \boldsymbol{\eta}^e|) < \tau_\theta$ . We vary the angle threshold  $\tau_\theta$  in the range  $[0^\circ, 5^\circ]$  with a step size of  $0.01^\circ$ . In Fig. 9, we show the precision rate vs the angle threshold  $\tau_\theta$  curves for different perturbation radius  $\sigma^2 \in \{0, 0.02, 0.04, \dots, 0.1\}$  and for  $d = 6$  and  $d = 8$ . We observe that the precision rate

increases as the angle threshold increases and decreases as the perturbation radius increases for both  $d = 6$  and  $d = 8$ .

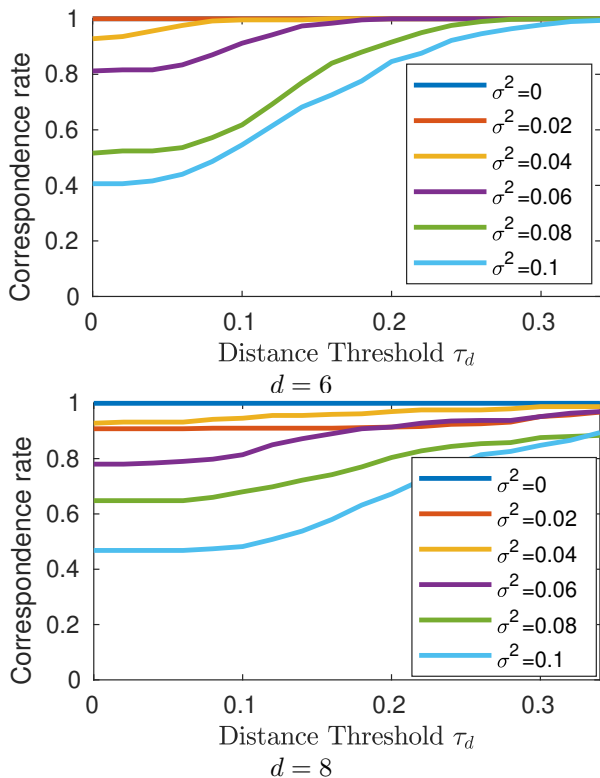


Fig. 8. Correspondence rate vs distance threshold curves for  $d = 6$  and  $d = 8$ .

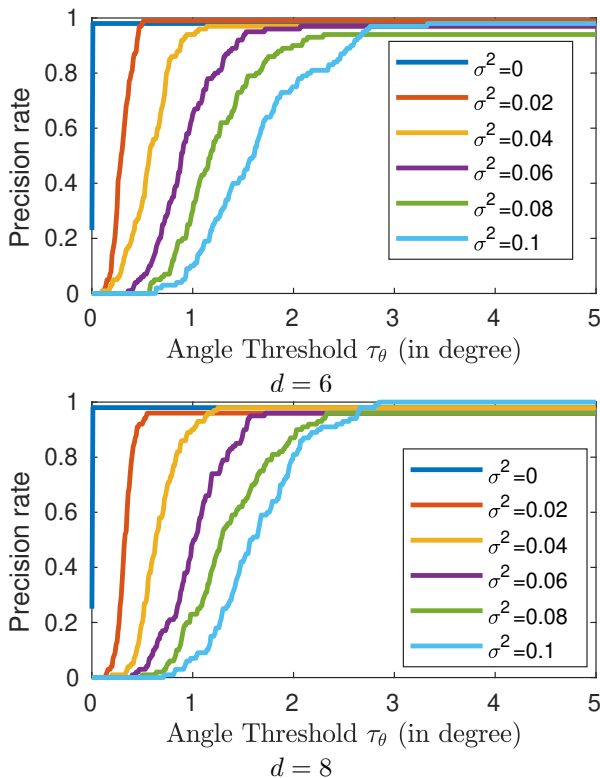


Fig. 9. Precision rate vs angle threshold curves for  $d = 6$  and  $d = 8$ .

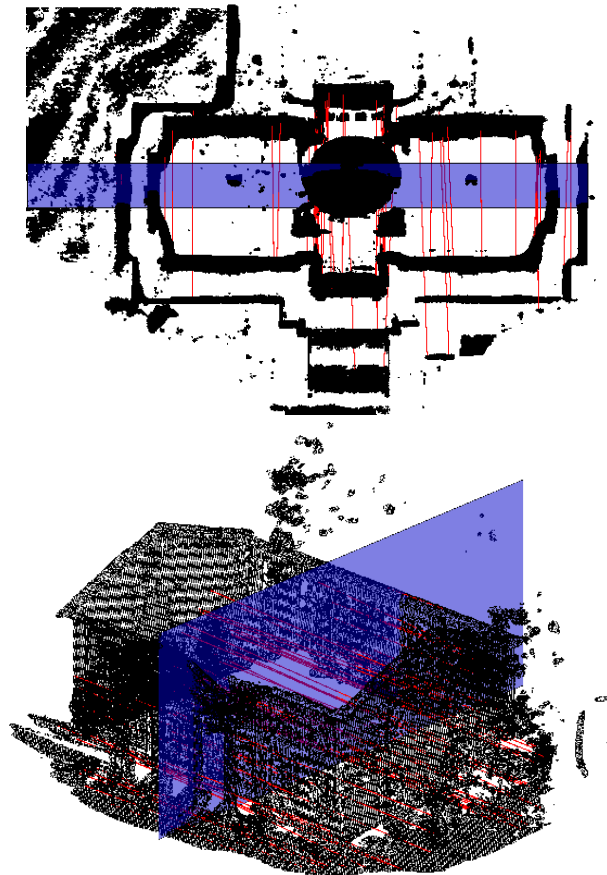


Fig. 10. Detected reflection symmetry on two real 3D scans of buildings from the dataset [56].

#### D. Results

In Fig. 10, we show the detected reflection symmetry for two real 3D scans of buildings from the dataset [56]. In Fig. 11, we present the results for the case  $d = 3$ . The point cloud in Fig. 10(a) contains 912045 points and the point cloud in Fig. 10(b) contains 767474 points. Since the computational complexity is  $O(n^{3.5}) + O(nd^2)$ , the computation time and space requirement (storing the matrices  $\mathbf{A}_1$  and  $\mathbf{A}_2$ ) are very high. Therefore, in order to compute the reflection symmetry in these scans, we randomly sample around 600 points. In both cases, we show the reflection symmetry plane by the blue color and estimated pairs of reflective symmetric points by the red colored line segment joining them. In order to make our algorithm robust to the part removal, we simply put the extra constraint  $\mathbf{e}^\top \mathbf{P} \mathbf{e} \leq 2k$  in ILP defined in Equation (17) which limits the number of pairs to at most  $k$ . For  $d = 2$ , we detect reflection symmetry in the set of corner points in a real image. In order to determine the symmetry axis, we use Theorem 2. For  $d = 3$ , we use existing standard 3D models dataset [57]. In order to calculate the symmetry axis in an image using the proposed approach, we first find the set of corner points [58]. This set may contain the corners not lying on the symmetric object. Therefore, we apply the proposed approach with RANSAC [59]. We compare the proposed results with the results of two descriptor based methods [39] and [46].

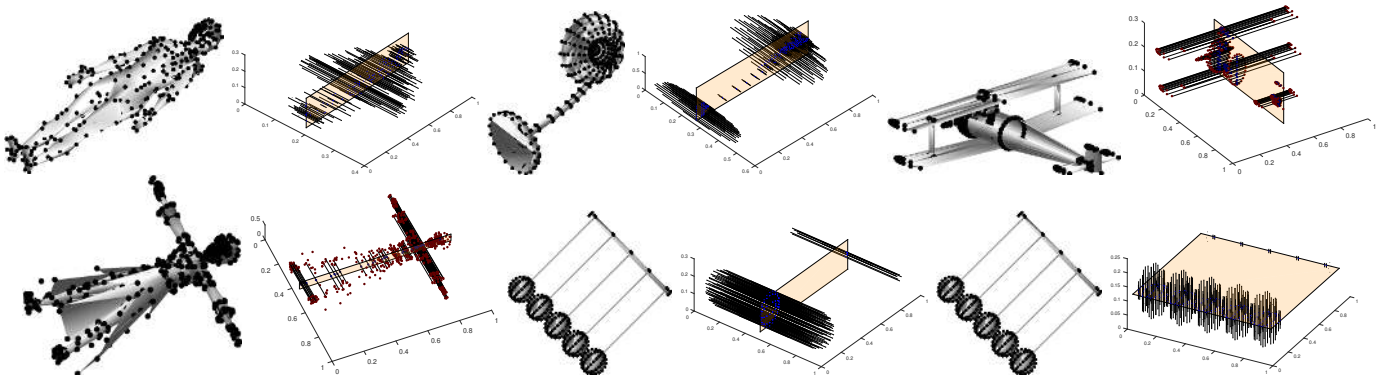


Fig. 11. Results of symmetry detection in the 3D object models from the dataset [57]. In the first and third columns we show the point set on the original surface. And in the second and fourth columns we show the detected reflection symmetry. The correspondences are shown by joining the mirror symmetric points by the black colored lines. The Reflection symmetry plane is shown in light brown color. The mid-points of the mirror symmetric points are show in blue color. Here, we show the surface for visualization purpose only.

We evaluate on real and synthetic images containing single symmetric object from the dataset [60]. In TABLE I, we present the precision and the recall rates. We observe that for synthetic images, the precision rate is very high for the proposed approach because most of the corner points lie on the symmetric object. Whereas, in real images, the set of corner points contains many outlier corners which leads to the degraded performance. Precision rates for the proposed approach are higher than that for the methods [39] and [46]. The recall rates are better than that of the method [39] and comparable to that of the method [46]. This leads to the conclusion that symmetry detection can be performed even when the feature descriptors are not available. In Fig. 12, we show the results on the datasets [61], [62], and [60]. The last two images show the failure cases from the datasets [60]. The reason could be that the pixels which are responsible for symmetry detection such as pixels on eyes and ear tips in the second image are not detected in the corner point detection step.

TABLE I  
PRECISION AND RECALL RATES FOR THE METHODS [39], [46], AND THE PROPOSED APPROACH ON THE DATASET [60].

	Precision			Recall		
	[39]	[46]	Ours	[39]	[46]	Ours
Real Images	0.21	0.30	<b>0.42</b>	0.75	<b>0.95</b>	0.93
Synthetic Images	0.28	0.29	<b>0.73</b>	0.93	<b>1.00</b>	0.96

**Influence of Different Initializations.** We first create the following dataset of 3D point clouds. We create 5000 point clouds  $\{S_i\}_{i=1}^{5000}$  with known ground-truth symmetries as discussed in §5.2. We keep 500 points in each point cloud. Without loss of generality, we choose the reflection symmetry plane such that it makes  $90^\circ$  angle with  $x$ -axis and  $y$ -axis, i.e., the  $x$ - $y$  plane. For each point cloud, we initialize the variable  $\mathbf{t}_i^0 = \text{mean}(S_i)$  and  $(\theta_x^0, \theta_y^0)$  on every point of the grid domain  $\{-90^\circ, -80^\circ, \dots, +80^\circ, +90^\circ\} \times \{-90^\circ, -80^\circ, \dots, +80^\circ, +90^\circ\}$ . We then run our approach and measure the error at the convergence  $e_i(\theta_x^0, \theta_y^0) = \|\mathbf{R}_x^* \mathbf{R}_y^* \mathbf{E}(\mathbf{R}_x^* \mathbf{R}_y^*)^\top (\mathbf{X}_i - \mathbf{t}^* \mathbf{e}^\top) + \mathbf{t}^* \mathbf{e}^\top - \mathbf{X}_i \mathbf{P}^*\|_F^2$  for each

initialization  $(\theta_x^0, \theta_y^0)$ . Then, we find the average error  $e(\theta_x^0, \theta_y^0) = \frac{1}{5000} \sum_{i=1}^{5000} e_i(\theta_x^0, \theta_y^0)$ . Here,  $\mathbf{R}_x$  and  $\mathbf{R}_y$  are defined as follows.

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x^0 & -\sin \theta_x^0 \\ 0 & \sin \theta_x^0 & \cos \theta_x^0 \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos \theta_y^0 & 0 & -\sin \theta_y^0 \\ 0 & 1 & 0 \\ \sin \theta_y^0 & 0 & \cos \theta_y^0 \end{bmatrix}.$$

In Fig. 13, we show the average error  $e(\theta_x^0, \theta_y^0)$ . We observe that if the initialization  $(\theta_x^0, \theta_y^0)$  is far away from the global optimum  $(0^\circ, 0^\circ)$ , then the error is very high. As the distance between the initialization angles  $(\theta_x^0, \theta_y^0)$  and the global optimum angles  $(0^\circ, 0^\circ)$  decrease, the error  $e(\theta_x^0, \theta_y^0)$  remains approximately constant and suddenly drops to near zero after a particular distance. This indicates that, if the initialization angles are within a particular distance from the global optimum, then our approach always find the global optimum solution. This empirical result concurs with the result we already proved in Theorems 5 and 6.

## VI. CONCLUSION

In this work, we have developed a theory for establishing the correspondences between the mirror symmetric points in  $\mathbb{R}^d$ . We, further, determine the reflection symmetry transformation in a volumetric set of points in  $\mathbb{R}^d$  containing a perturbed reflection symmetry pattern using optimization on Riemannian manifold. We have shown that our method is robust to a significant amount of perturbation and achieves 100% accuracy for no perturbation. We have further shown the significance of this work by detecting reflection symmetry in real images and comparing with state-of-the-art methods. The proposed approach is particularly suitable for detecting reflection symmetry of objects in applications where obtaining a robust local descriptor is highly challenging. The linear assignment problem is a time consuming step which restricts us to apply it on the large point sets. However, a proper sampling method can be employed to reduce the size of the point set without losing the symmetry present in the point set. We believe that the fundamental theory and algorithm developed in this work will pave the way for researchers to exploit them for scenarios where estimating feature descriptors is a challenging task.

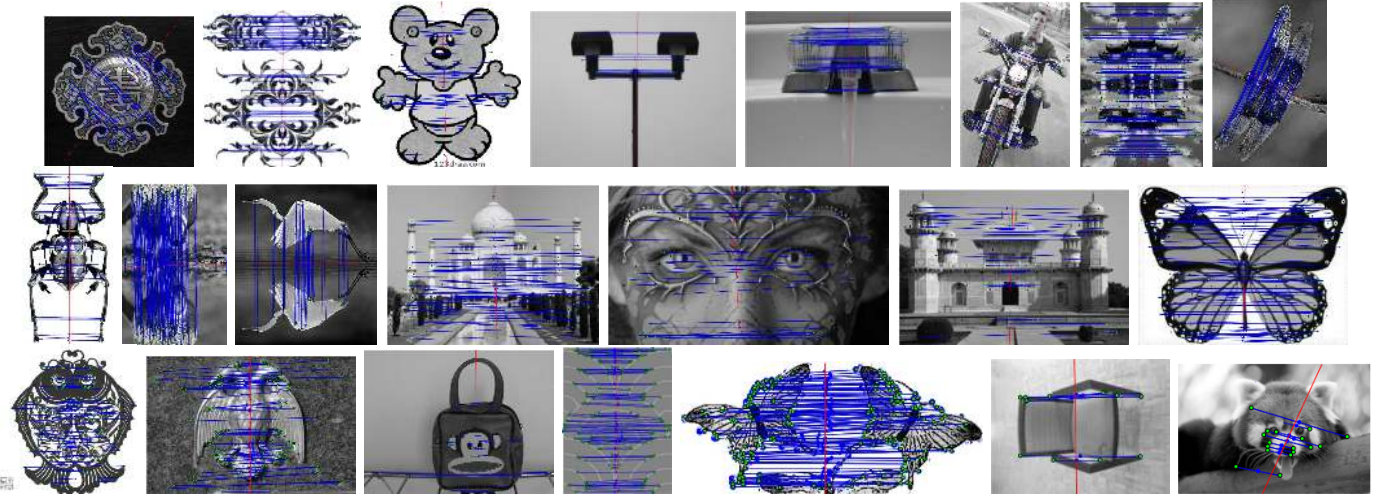


Fig. 12. Results of symmetry detection in real images from the dataset [62], [61], [60]. We show the set  $\mathcal{S}$  using green points, the reflection symmetry axis by a red line, and the correspondences between the mirror symmetric points by the blue lines.

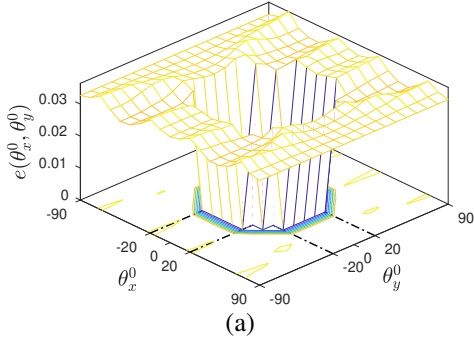


Fig. 13. Average error  $e(\theta_x^0, \theta_y^0)$  vs the initialization angles  $(\theta_x^0, \theta_y^0)$ .

Our approach detects single reflection symmetry plane of an object. Consider the third row of Fig. 11 in which there are multiple reflection symmetry planes present. In such cases, the detected reflection symmetry plane will be the one to which the initialized plane is the closest. For example, in the third row of Fig. 11, we have shown both the reflection symmetry planes detected depending on different initializations. This may not be a proper way of detecting multiple symmetries, though this is an interesting direction. We would like to extend our approach for the detection of multiple reflection symmetry planes of a symmetric object exhibiting multiple symmetries or a point cloud containing more than one symmetric objects.

## VII. APPENDIX

### A1. Euclidean gradient of the function $\bar{f}$ with respect to the variable $\mathbf{t}$ (Equation (10))

We write the cost function as follows.

$$\begin{aligned}\bar{f}(\mathbf{R}, \mathbf{t}, \mathbf{P}) &= \|\mathbf{TET}^\top(\mathbf{X} - \mathbf{te}^\top) - (\mathbf{XP} - \mathbf{te}^\top)\|_{\mathbb{F}}^2 \\ &= \|(\mathbf{TET}^\top\mathbf{X} - \mathbf{XP}) + (\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top\|_{\mathbb{F}}^2.\end{aligned}$$

We note that

$$(\mathbf{I}_d - \mathbf{TET}^\top)^\top(\mathbf{I}_d - \mathbf{TET}^\top) = 2(\mathbf{I}_d - \mathbf{TET}^\top).$$

Therefore, we have (the terms which are not functions of  $\mathbf{t}$  are not shown)

$$\begin{aligned}\bar{f}(\mathbf{R}, \mathbf{t}, \mathbf{P}) &= \text{trace}(2\mathbf{te}^\top(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top \\ &\quad + 2(\mathbf{X}^\top\mathbf{TET}^\top - \mathbf{P}^\top\mathbf{X}^\top)(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top).\end{aligned}$$

Now taking the derivative with respect to  $\mathbf{t}$  we have,

$$\begin{aligned}\nabla_{\mathbf{t}}\bar{f} &= 2(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top\mathbf{e} + 2(\mathbf{e}^\top\mathbf{e}\mathbf{t}^\top(\mathbf{I}_d - \mathbf{TET}^\top))^\top \\ &\quad + 2(\mathbf{e}^\top(\mathbf{X}^\top\mathbf{TET}^\top - \mathbf{P}^\top\mathbf{X}^\top)(\mathbf{I}_d - \mathbf{TET}^\top))^\top \\ &= 4(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top\mathbf{e} + 2(\mathbf{I}_d - \mathbf{TET}^\top)(\mathbf{TET}^\top\mathbf{X} - \mathbf{XP})\mathbf{e}\end{aligned}$$

Here we have that  $(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{TET}^\top = -(\mathbf{I}_d - \mathbf{TET}^\top)$ . Therefore,

$$\begin{aligned}\nabla_{\mathbf{t}}\bar{f} &= 4(\mathbf{I}_d - \mathbf{TET}^\top)\mathbf{te}^\top\mathbf{e} - 2(\mathbf{I}_d - \mathbf{TET}^\top)(\mathbf{X} + \mathbf{XP})\mathbf{e} \\ &= 2(\mathbf{I}_d - \mathbf{TET}^\top)(2\mathbf{te}^\top\mathbf{e} - \mathbf{X}\mathbf{e} - \mathbf{XP}\mathbf{e}).\end{aligned}$$

### A2. Euclidean gradient of the function $\bar{f}$ with respect to the variable $\mathbf{R}_j$ (Equation (11))

Let us consider the cost function as defined in equation (5) (in main manuscript):

$$\bar{f}(\mathbf{R}, \mathbf{t}, \mathbf{P}) = \left\| \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top (\mathbf{X} - \mathbf{te}^\top) + \mathbf{te}^\top - \mathbf{XP} \right\|_{\mathbb{F}}^2.$$

Now, let us define  $\mathbf{T} = \prod_{u=1}^{d-1} \mathbf{R}_u$ ,  $\mathbf{U} = \mathbf{X} - \mathbf{te}^\top$  and  $\mathbf{V} = \mathbf{XP} - \mathbf{te}^\top$ .

Then the cost function becomes.

$$\begin{aligned}\bar{f}(\mathbf{R}, \mathbf{t}, \mathbf{P}) &= \|\mathbf{TET}^\top\mathbf{U} - \mathbf{V}\|_{\mathbb{F}}^2 \\ &= \text{trace}((\mathbf{TET}^\top\mathbf{U} - \mathbf{V})^\top(\mathbf{TET}^\top\mathbf{U} - \mathbf{V})) \\ &= \text{trace}((\mathbf{U}^\top\mathbf{TET}^\top - \mathbf{V}^\top)(\mathbf{TET}^\top\mathbf{U} - \mathbf{V})) \\ &= \text{trace}(\mathbf{U}^\top\mathbf{TET}^\top\mathbf{TET}^\top\mathbf{U} - 2\mathbf{U}^\top\mathbf{TET}^\top\mathbf{V} \\ &\quad + \mathbf{V}^\top\mathbf{V})\end{aligned}$$

Here we note that  $\mathbf{TET}^\top\mathbf{TET}^\top = \mathbf{I}_d$ , therefore

$$\bar{f}(\mathbf{R}, \mathbf{t}, \mathbf{P}) = \text{trace}(\mathbf{U}^\top\mathbf{U} - 2\mathbf{U}^\top\mathbf{TET}^\top\mathbf{V} + \mathbf{V}^\top\mathbf{V}).$$

Now taking the classical gradient of  $\bar{f}$  with respect to  $\mathbf{R}_j$  we have. (We follow [63] for the necessary properties.)

$$\begin{aligned} \frac{\partial \bar{f}}{\partial \mathbf{R}_j} &= -2 \frac{\partial}{\partial \mathbf{R}_j} \text{trace}(\mathbf{U}^\top \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{V}) \\ &= -2 \frac{\partial}{\partial \mathbf{R}_j} \text{trace}(\mathbf{U}^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right) \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \\ &\quad \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{R}_j^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{V}) \\ &= -2 \left( \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{V} \mathbf{U}^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right) \right)^\top \\ &\quad - 2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{V} \mathbf{U}^\top \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &= -2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{U} \mathbf{V}^\top \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &\quad - 2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{V} \mathbf{U}^\top \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &= -2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top (\mathbf{U} \mathbf{V}^\top + \mathbf{V} \mathbf{U}^\top) \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &= -2 \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \end{aligned}$$

Where

$$\begin{aligned} \mathbf{A} &= (\mathbf{V} \mathbf{U}^\top + \mathbf{U} \mathbf{V}^\top) \\ &= (\mathbf{X} \mathbf{P} - \mathbf{t} \mathbf{e}^\top)(\mathbf{X} - \mathbf{t} \mathbf{e}^\top)^\top + (\mathbf{X} - \mathbf{t} \mathbf{e}^\top)(\mathbf{X} \mathbf{P} - \mathbf{t} \mathbf{e}^\top)^\top. = -\mathbf{R}_j \mathbf{R}_j^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right) \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &\quad + \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{R}_j^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right) \mathbf{R}_j \\ &= -\mathbf{R}_j \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j \mathbf{B}_1 + \mathbf{R}_j \mathbf{B}_1 \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j. \end{aligned}$$

A3. The Riemannian gradient of the function  $f$  with respect to the variable  $\mathbf{t}$  (Equation (12))

Using the definition, as defined in main paper, of Riemannian gradient  $\xi_{\mathbf{t}}(\mathbf{t})$  of the function  $f$  with respect to the variable  $\mathbf{t}$  we have

$$\begin{aligned} \xi_{\mathbf{t}}(\mathbf{t}) &= \mathbb{P}_{\mathbf{t}}(\nabla_{\mathbf{t}} \bar{f}) = \nabla_{\mathbf{t}} \bar{f} \\ &= 2(\mathbf{I}_d - \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top)(2\mathbf{n} \mathbf{t} - \mathbf{X} \mathbf{e} - \mathbf{X} \mathbf{P} \mathbf{e}). \end{aligned}$$

A4. The Riemannian gradient of the function  $f$  with respect to the variable  $\mathbf{R}_j$  (Equation (13))

Using the definition, as defined in main paper, of Riemannian gradient  $\xi_{\mathbf{R}_j}(\mathbf{R}_j)$  of the function  $f$  with respect to the variable  $\mathbf{R}_j$  we have

$$\begin{aligned} \xi_{\mathbf{R}_j}(\mathbf{R}_j) &= \mathbb{P}_{\mathbf{R}_j}(\nabla_{\mathbf{R}_j} \bar{f}) = \mathbf{R}_j \text{skew}(\mathbf{R}_j^\top \nabla_{\mathbf{R}_j} \bar{f}). \\ \xi_{\mathbf{R}_j}(\mathbf{R}_j) &= \mathbf{R}_j \text{skew}(\mathbf{R}_j^\top \nabla_{\mathbf{R}_j} \bar{f}). \\ \mathbf{R}_j^\top \nabla_{\mathbf{R}_j} \bar{f} &= -2 \left( \prod_{u=1}^j \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ \nabla_{\mathbf{R}_j} \bar{f}^\top \mathbf{R}_j &= -2 \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^j \mathbf{R}_u \right). \end{aligned}$$

Therefore,

$$\begin{aligned} \xi_{\mathbf{R}_j}(\mathbf{R}_j) &= \mathbf{R}_j \frac{\mathbf{R}_j^\top \nabla_{\mathbf{R}_j} \bar{f} - \nabla_{\mathbf{R}_j} \bar{f}^\top \mathbf{R}_j}{2} \\ &= -\mathbf{R}_j \left( \prod_{u=1}^j \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &\quad + \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^j \mathbf{R}_u \right). \quad (19) \end{aligned}$$

A5. The Riemannian Hessian of the function  $f$  with respect to  $\mathbf{R}_j$  (Equation (14))

Next, we determine the Riemannian Hessian of the function  $f$ . In order to determine the  $j^{\text{th}}$  component  $\text{Hess}_{\mathbf{R}_j}(f(\mathbf{R}, \mathbf{t}))[\mathbf{R}_j \Omega_j] = \mathbb{P}_{\mathbf{R}_j}(\mathbf{D} \xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j \Omega_j])$ , of the Riemannian Hessian, we first find the classical derivative  $\mathbf{D} \xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j \Omega_j]$  of the Riemannian gradient  $\xi_{\mathbf{R}_j}(\mathbf{R}_j)$  in the direction  $\mathbf{R}_j \Omega_j$  and then we apply the projection operator  $\mathbb{P}_{\mathbf{R}_j}$ . Now using Equation 19 we have

$$\begin{aligned} \xi_{\mathbf{R}_j}(\mathbf{R}_j) &= -\mathbf{R}_j \left( \prod_{u=1}^j \mathbf{R}_u \right)^\top \mathbf{A} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \\ &\quad + \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^j \mathbf{R}_u \right) \\ &\quad + \mathbf{R}_j \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top \mathbf{R}_j^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right) \mathbf{R}_j \\ &= -\mathbf{R}_j \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j \mathbf{B}_1 + \mathbf{R}_j \mathbf{B}_1 \mathbf{R}_j^\top \mathbf{B}_2 \mathbf{R}_j. \end{aligned}$$

Here,

$$\begin{aligned} \mathbf{B}_1 &= \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right) \mathbf{E} \left( \prod_{u=j+1}^{d-1} \mathbf{R}_u \right)^\top, \\ \mathbf{B}_2 &= \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right)^\top \mathbf{A}^\top \left( \prod_{u=1}^{j-1} \mathbf{R}_u \right). \end{aligned}$$

Now

$$\begin{aligned} D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j] &= \frac{d}{dt}\xi_{\mathbf{R}_j}(\mathbf{R}_j + t\mathbf{R}_j\Omega_j) \Big|_{t=0} \\ &= \frac{d}{dt}(-(\mathbf{R}_j + t\mathbf{R}_j\Omega_j)(\mathbf{R}_j + t\mathbf{R}_j\Omega_j)^\top \mathbf{B}_2(\mathbf{R}_j + t\mathbf{R}_j\Omega_j)\mathbf{B}_1) \Big|_{t=0} \\ &+ \frac{d}{dt}((\mathbf{R}_j + t\mathbf{R}_j\Omega_j)\mathbf{B}_1(\mathbf{R}_j + t\mathbf{R}_j\Omega_j)^\top \mathbf{B}_2(\mathbf{R}_j + t\mathbf{R}_j\Omega_j)) \Big|_{t=0}. \end{aligned}$$

The first term is equal to

$$-\mathbf{R}_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1.$$

The second term is equal to

$$\mathbf{R}_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j - \mathbf{R}_j\mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j + \mathbf{R}_j\Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j.$$

Therefore,

$$\begin{aligned} D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j] &= -\mathbf{R}_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 + (\mathbf{R}_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j \\ &- \mathbf{R}_j\mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j + \mathbf{R}_j\Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j)\mathbf{R}_j^\top D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j] \\ &= -\mathbf{R}_j^\top \mathbf{R}_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 + \mathbf{R}_j^\top (\mathbf{R}_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j \\ &- \mathbf{R}_j\mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j + \mathbf{R}_j\Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j) \\ &= -\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 + (\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j \\ &- \mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j + \Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j)(D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j])^\top \mathbf{R}_j \\ &= \mathbf{B}_1^\top \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{R}_j^\top \mathbf{R}_j + (-\Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \mathbf{R}_j^\top \\ &+ \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\Omega_j\mathbf{B}_1^\top \mathbf{R}_j^\top - \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \Omega_j\mathbf{R}_j^\top)\mathbf{R}_j \\ &= \mathbf{B}_1^\top \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j + (-\Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \\ &+ \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\Omega_j\mathbf{B}_1^\top - \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \Omega_j) \end{aligned}$$

$$\begin{aligned} &\mathbf{R}_j^\top D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j] - (D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j])^\top \mathbf{R}_j \\ &= -\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 + \mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j - \mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j \\ &+ \Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j - \mathbf{B}_1^\top \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j + \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \\ &- \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\Omega_j\mathbf{B}_1^\top + \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \Omega_j \\ &= \mathbf{B}_1[\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j, \Omega_j] - [\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j, \Omega_j]\mathbf{B}_1 + \\ &[\Omega_j, \mathbf{B}_1]\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j - \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j[\Omega_j, \mathbf{B}_1] \\ &= [\mathbf{B}_1, [\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j, \Omega_j]] + [[\Omega_j, \mathbf{B}_1], \mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j]. \end{aligned}$$

Here  $[\cdot, \cdot]$  is the Lie bracket and defined as  $[\mathbf{U}, \mathbf{V}] = \mathbf{UV} - \mathbf{VU}$  for any two matrices  $\mathbf{U}$  and  $\mathbf{V}$ .

$$\begin{aligned} \text{Hess}_{\mathbf{R}_j}(f(\mathbf{R}, \mathbf{t}))[\mathbf{R}_j\Omega_j] &= \mathbb{P}_{\mathbf{R}_j}(D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j]) \\ &= \mathbf{R}_j \text{skew}(\mathbf{R}_j^\top D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j]) \\ &= \frac{1}{2}\mathbf{R}_j(\mathbf{R}_j^\top D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j] - (D\xi_{\mathbf{R}_j}(\mathbf{R}_j)[\mathbf{R}_j\Omega_j])^\top \mathbf{R}_j) \\ &= \frac{1}{2}\mathbf{R}_j([\mathbf{B}_1, [\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j, \Omega_j]] + [[\Omega_j, \mathbf{B}_1], \mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j]). \end{aligned}$$

A6. The Riemannian Hessian of the function  $f$  with respect to  $\mathbf{t}$  (Equation (15))

In a similar way, we determine the second component,  $\mathbb{P}_{\mathbf{t}}(D\xi_{\mathbf{t}}(\mathbf{t})[\eta_{\mathbf{t}}])$ , of the Riemannian Hessian. Since  $\mathbb{R}^d$  is a vector space we have  $\mathbb{P}_{\mathbf{t}}(D\xi_{\mathbf{t}}(\mathbf{t})[\eta_{\mathbf{t}}]) = D\xi_{\mathbf{t}}(\mathbf{t})[\eta_{\mathbf{t}}]$

$$D\xi_{\mathbf{t}}(\mathbf{t})[\eta_{\mathbf{t}}] = \frac{d}{dq}\xi_{\mathbf{t}}(\mathbf{t} + q\eta_{\mathbf{t}}) \Big|_{q=0}$$

$$= 4n(\mathbf{I}_d - \left(\prod_{u=1}^{d-1} \mathbf{R}_u\right)\mathbf{E}\left(\prod_{u=1}^{d-1} \mathbf{R}_u\right)^\top)\eta_{\mathbf{t}}.$$

Therefore

$$\text{Hess}_{\mathbf{t}}(f(\mathbf{R}, \mathbf{t}))[\eta_{\mathbf{t}}] = 4n(\mathbf{I}_d - \left(\prod_{u=1}^{d-1} \mathbf{R}_u\right)\mathbf{E}\left(\prod_{u=1}^{d-1} \mathbf{R}_u\right)^\top)\eta_{\mathbf{t}}. \quad (20)$$

A7. Steps of Theorem 5

Showing the fact

$$\text{trace}(\Omega_j^\top \mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j\Omega_j]) = 4\text{trace}(\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j(\mathbf{B}_1\Omega_j - \Omega_j\mathbf{B}_1)).$$

Now

$$\begin{aligned} \text{trace}(\Omega_j^\top \mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j\Omega_j]) &= \text{trace}(-\Omega_j^\top \mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 \\ &+ \Omega_j^\top \mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j - \Omega_j^\top \mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j + \Omega_j^\top \Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j \\ &- \Omega_j^\top \mathbf{B}_1^\top \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j + \Omega_j^\top \Omega_j\mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \\ &- \Omega_j^\top \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\Omega_j\mathbf{B}_1^\top + \Omega_j^\top \mathbf{R}_j^\top \mathbf{B}_2^\top \mathbf{R}_j\mathbf{B}_1^\top \Omega_j) \\ &= \text{trace}(\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 - \Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j \\ &+ \Omega_j\mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j - \Omega_j\Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j \\ &+ \Omega_j\mathbf{B}_1\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j - \Omega_j\Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\mathbf{B}_1 \\ &+ \Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1 - \Omega_j\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\mathbf{B}_1\Omega_j) \\ &= \text{trace}(4\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1\Omega_j - 2\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\Omega_j\mathbf{B}_1 \\ &- 2\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\mathbf{B}_1\Omega_j\Omega_j) \end{aligned}$$

Since,

$$\begin{aligned} \text{trace}(\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\mathbf{B}_1\Omega_j\Omega_j) &= \text{trace}((\mathbf{B}_1\Omega_j\Omega_j)^\top \mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j) \\ &= \text{trace}(\Omega_j\Omega_j\mathbf{B}_1\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j) \\ &= \text{trace}(\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\Omega_j\mathbf{B}_1), \end{aligned}$$

we have

$$\begin{aligned} \text{trace}(\Omega_j^\top \mathbf{R}_j^\top \mathbf{H}[\mathbf{R}_j\Omega_j]) &= \text{trace}(4\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\mathbf{B}_1\Omega_j \\ &- 4\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j\Omega_j\Omega_j\mathbf{B}_1) \\ &= 4\text{trace}(\mathbf{R}_j^\top \mathbf{B}_2\mathbf{R}_j(\Omega_j\mathbf{B}_1\Omega_j - \Omega_j\Omega_j\mathbf{B}_1)). \end{aligned}$$

## REFERENCES

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *Computer Graphics Forum*, vol. 32, no. 6. Wiley Online Library, 2013, pp. 1–23.
- [3] A. Berner, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel, "A graph-based approach to symmetry detection," in *Volume Graphics*, vol. 40, 2008, pp. 1–8.
- [4] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 560–568.
- [5] M. Ovsjanikov, J. Sun, and L. Guibas, "Global intrinsic symmetries of shapes," in *Computer graphics forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1341–1348.
- [6] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, "Symmetry factored embedding and distance," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 103.
- [7] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu, and B. Chen, "Multi-scale partial intrinsic symmetry detection," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 181, 2012.
- [8] Y. Liu, H. Hel-Or, and C. S. Kaplan, *Computational symmetry in computer vision and computer graphics*. Foundations and Trends in Computer Vision and Computer Graphics, Now publishers Inc, 2010.
- [9] H. Zabrodsky, S. Peleg, and D. Avnir, "Symmetry as a continuous feature," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1154–1166, 1995.
- [10] B. Combès, R. Hennessy, J. Waddington, N. Roberts, and S. Prima, "Automatic symmetry plane estimation of bilateral objects in point clouds," in *IEEE CVPR*. IEEE, 2008, pp. 1–8.
- [11] P. Speciale, M. R. Oswald, A. Cohen, and M. Pollefeys, "A symmetry prior for convex variational 3d reconstruction," in *ECCV*. Springer, 2016, pp. 313–328.
- [12] A. Ecin, C. Fermuller, and Y. Aloimonos, "Detecting reflectional symmetries in 3d data through symmetrical fitting," in *IEEE ICCV Workshop on Detecting Symmetry in the Wild, Venice*, vol. 7, 2017, p. 8.
- [13] M. Cicconet, D. G. C. Hildebrand, and H. Elliott, "Finding mirror symmetry via registration and optimal symmetric pairwise assignment of curves," in *IEEE ICCV Workshops*, Oct 2017.
- [14] B. Li, H. Johan, Y. Ye, and Y. Lu, "Efficient 3d reflection symmetry detection: A view-based approach," *Graphical Models*, vol. 83, pp. 2–14, 2016.
- [15] I. Sipiran, R. Gregor, and T. Schreck, "Approximate symmetry detection in partial 3d meshes," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 131–140.
- [16] A. Martinet, C. Soler, N. Holzschuch, and F. X. Sillion, "Accurate detection of symmetries in 3d shapes," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 2, pp. 439–464, 2006.
- [17] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Symmetries of non-rigid shapes," in *ICCV*. IEEE, 2007.
- [18] A. Cohen, C. Zach, S. N. Sinha, and M. Pollefeys, "Discovering and exploiting 3d symmetries in structure from motion," in *IEEE CVPR*. IEEE, 2012, pp. 1514–1521.
- [19] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, and A. Schilling, "Symmetry detection using feature lines," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 697–706.
- [20] R. Lasowski, A. Tevs, H.-P. Seidel, and M. Wand, "A probabilistic framework for partial intrinsic symmetries in geometric data," in *IEEE ICCV*. IEEE, 2009, pp. 963–970.
- [21] N. J. Mitra, L. J. Guibas, and M. Pauly, "Symmetrization," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 63, 2007.
- [22] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 43.
- [23] D. M. Thomas and V. Natarajan, "Multiscale symmetry detection in scalar fields by clustering contours," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2427–2436, 2014.
- [24] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser, "A reflective symmetry descriptor," *ECCV*, pp. 777–778, 2002.
- [25] H. Liu, J. Xia, J. Chen, and J. Wang, "Detection of hierarchical intrinsic symmetry structure in 3d models," *Computers & Graphics*, 2017.
- [26] Z. Shi, P. Alliez, M. Desbrun, H. Bao, and J. Huang, "Symmetry and orbit detection via lie-algebra voting," in *Computer Graphics Forum*, vol. 35, no. 5. Wiley Online Library, 2016, pp. 217–227.
- [27] H. Wang and H. Huang, "Group representation of global intrinsic symmetries," in *Computer Graphics Forum*, vol. 36, no. 7. Wiley Online Library, 2017, pp. 51–61.
- [28] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser, "Möbius transformations for global intrinsic symmetry analysis," in *Computer Graphics Forum*, vol. 29, no. 5. Wiley Online Library, 2010, pp. 1689–1700.
- [29] N. J. Mitra, A. Bronstein, and M. Bronstein, "Intrinsic regularity detection in 3d geometry," in *ECCV*. Springer, 2010, pp. 398–410.
- [30] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3d shapes," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 549–559, 2006.
- [31] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Full and partial symmetries of non-rigid shapes," *International Journal of Computer Vision*, vol. 89, no. 1, pp. 18–39, 2010.
- [32] S. Thrun and B. Wegbreit, "Shape from symmetry," in *IEEE ICCV*, vol. 2. IEEE, 2005, pp. 1824–1831.
- [33] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, and Y. Xiong, "Partial intrinsic reflectional symmetry of 3d shapes," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5. ACM, 2009, p. 138.
- [34] C. Sun and J. Sherrah, "3d symmetry detection using the extended gaussian image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 164–168, 1997.
- [35] W. Jiang, K. Xu, Z.-Q. Cheng, and H. Zhang, "Skeleton-based intrinsic symmetry detection on point clouds," *Graphical Models*, vol. 75, no. 4, pp. 177–188, 2013.
- [36] M. Lukáč, D. Sýkora, K. Sunkavalli, E. Shechtman, O. Jamriška, N. Carr, and T. Pajdla, "Nautilus: Recovering regional symmetry transformations for image editing," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.
- [37] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Multiscale symmetric part detection and grouping," *International journal of computer vision*, vol. 104, no. 2, pp. 117–134, 2013.
- [38] T. Sie Ho Lee, S. Fidler, and S. Dickinson, "Detecting curved symmetric parts using a deformable disc model," in *ICCV*, 2013, pp. 1753–1760.
- [39] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *ECCV*. Springer, 2006, pp. 508–521.
- [40] C. L. Teo, C. Fermuller, and Y. Aloimonos, "Detection and segmentation of 2d curved reflection symmetric structures," in *IEEE ICCV*, 2015, pp. 1644–1652.
- [41] D. C. Hauage and N. Snavely, "Image matching using local symmetry features," in *IEEE CVPR*. IEEE, 2012, pp. 206–213.
- [42] Z. Wang, Z. Tang, and X. Zhang, "Reflection symmetry detection using locally affine invariant edge correspondence," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1297–1301, 2015.
- [43] V. Patraucean, R. Grompone von Gioi, and M. Ovsjanikov, "Detection of mirror-symmetric image patches," in *IEEE CVPR Workshops*, 2013, pp. 211–216.
- [44] S. Tsogkas and I. Kokkinos, "Learning-based symmetry detection in natural images," in *ECCV*. Springer, 2012, pp. 41–54.
- [45] M. Chertok and Y. Keller, "Spectral symmetry analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1227–1238, 2010.
- [46] I. R. Atadjanov and S. Lee, "Reflection symmetry detection via appearance of structure descriptor," in *ECCV*. Springer, 2016, pp. 3–18.
- [47] K. Köser, C. Zach, and M. Pollefeys, "Dense 3d reconstruction of symmetric scenes from a single image," in *Joint Pattern Recognition Symposium*. Springer, 2011, pp. 266–275.
- [48] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.
- [49] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.
- [50] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [51] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.
- [52] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014. [Online]. Available: <http://www.manopt.org>
- [53] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984, pp. 302–311.
- [54] J. Kleinberg and E. Tardos, *Algorithm design*. Pearson Education India, 2006.



- [55] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.
- [56] C. Funk, S. Lee, M. R. Oswald, S. Tsogkas, W. Shen, A. Cohen, S. Dickinson, and Y. Liu, "2017 iccv challenge: Detecting symmetry in the wild," in *IEEE ICCV Workshops*, Oct 2017.
- [57] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Shape modeling applications, 2004. Proceedings. IEEE*, 2004, pp. 167–178.
- [58] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Manchester, UK, 1988, pp. 10–5244.
- [59] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [60] I. Rauschert, J. Liu, and Y. Liu, "Symmetry detection competition: A summary of how the competition is carried out," in *IEEE CVPR Workshop Symmetry Detection Real World Images*, 2011, pp. 1–66.
- [61] M. Cicconet, V. Birodkar, M. Lund, M. Werman, and D. Geiger, "A convolutional approach to reflection symmetry," *arXiv preprint arXiv:1609.05257*, 2016.
- [62] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu, "Symmetry detection from realworld images competition 2013: Summary and results," in *IEEE CVPR Workshops*, 2013, pp. 200–205.
- [63] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.