

# A Solution for Multi-Alignment by Transformation Synchronisation

Florian Bernard<sup>1,2,3</sup>

Johan Thunberg<sup>2</sup>

Peter Gemmar<sup>3</sup>

Frank Hertel<sup>1</sup>

Andreas Husch<sup>1,2,3</sup>

Jorge Goncalves<sup>2</sup>

<sup>1</sup>Centre Hospitalier de Luxembourg, Luxembourg

{bernard.florian,hertel.frank,husch.andreas}@chl.lu

<sup>2</sup>Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Luxembourg

{johan.thunberg,jorge.goncalves}@uni.lu

<sup>3</sup>Trier University of Applied Sciences, Germany

p.gemmar@hochschule-trier.de

## Abstract

*The alignment of a set of objects by means of transformations plays an important role in computer vision. Whilst the case for only two objects can be solved globally, when multiple objects are considered usually iterative methods are used. In practice the iterative methods perform well if the relative transformations between any pair of objects are free of noise. However, if only noisy relative transformations are available (e.g. due to missing data or wrong correspondences) the iterative methods may fail.*

*Based on the observation that the underlying noise-free transformations can be retrieved from the null space of a matrix that can directly be obtained from pairwise alignments, this paper presents a novel method for the synchronisation of pairwise transformations such that they are transitively consistent.*

*Simulations demonstrate that for noisy transformations, a large proportion of missing data and even for wrong correspondence assignments the method delivers encouraging results.*

## 1. Introduction

The alignment of a set of objects by means of transformations plays an important role in the field of computer vision and recognition. For instance, for the creation of statistical shape models (SSMs) [5] training shapes are initially aligned for removing pose differences in order to only model shape variability.

The most common way of shape representation is by encoding each shape as a point-cloud. In order to be able to process a set of shapes it is necessary that correspondences

between all shapes are established. Whilst there is a vast amount of research in the field of shape correspondences (for an overview see [11, 18]), in this paper we focus on the alignment of shapes and we assume that correspondences have already been established.

The alignment of two objects by removing location, scale and rotation is known as *Absolute Orientation Problem (AOP)* [13] or *Procrustes Analysis* [8]. For the AOP there are various closed-form solutions, among them methods based on singular value decomposition (SVD) [1, 16]; based on eigenvalue decomposition [13]; based on unit quaternions [12] or based on dual quaternions [19]. A comparison of these methods [6] has revealed that the accuracy and the robustness of all methods are comparable.

The alignment of more than two objects is known as *Generalised Procrustes Analysis (GPA)*. Whilst a computationally expensive global solution for GPA in two and three dimensions has been presented in [15], the most common way for solving the GPA is to align the objects with a reference object. However, fixing any of the objects as reference induces a bias. An unbiased alternative is to align all objects with the adaptive mean object as reference. An iterative algorithm then alternately updates the reference object and estimates the transformations aligning the objects. The iterative nature of these methods constitutes a problem if the relative transformation between any pair of objects is noisy. This is for example the case if data is missing, correspondences are wrong or if the transformations are observed by independent sensors (e.g. non-communicating robots observe each other). Noisy relative transformations can be characterised by transitive inconsistency, i.e. transforming  $A$  to  $B$  and  $B$  to  $C$  might lead to a different result than transforming  $A$  directly to  $C$ .

This paper presents a novel method for synchronising

the set of all pairwise transformations in such a way that they globally exhibit transitive consistency. Experiments demonstrate the effectiveness of this method in denoising noisy pairwise transformations. Furthermore, using this novel method the GPA is solved in an unbiased manner in closed-form, *i.e.* non-iterative. Transformation synchronisation is applied to solve the GPA with missing data as well as with wrong correspondence assignments and results in superior performance compared to existing methods.

Our main contribution is a generalisation of the techniques presented by Singer *et al.* [3, 9, 10, 17], who have introduced a method for minimising global self-consistency errors between pairwise orthogonal transformations based on eigenvalue decomposition and semidefinite programming. With permutation transformations being a subset of orthogonal transformations, in [14] the authors demonstrate that the method by Singer *et al.* is also able to effectively synchronise permutation transformations for globally consistent matchings.

In our case, rather than considering the special case of orthogonal matrices, we present a synchronisation method for invertible linear transformations. Furthermore, it is demonstrated how this method can be applied for the synchronisation of similarity, euclidean and rigid transformations, which are of special interest for the groupwise alignment of shapes.

Whilst the proposed synchronisation method is applicable in many other fields where noisy pairwise transformations are to be denoised (*e.g.* groupwise image registration or multi-view registration), in this paper GPA is used as illustrating example.

## 2. Methods

For the presentation of our novel transformation synchronisation method the notation and some foundations are introduced first. Subsequently, a formulation for the case of perfect information is given. Motivated by these elaborations, a straightforward extension to handle noisy pairwise transformations is presented. Finally, various types of transformations are discussed.

### 2.1. Notation and Foundations

$\mathbf{X}_i, \mathbf{X}_j \in \mathbb{R}^{n \times d}$  are matrices representing point-clouds with  $n$  points in  $d$  dimensions where in the following all  $\mathbf{X}_i$  are simply referred to as point-clouds. Let  $\mathbf{I}$  be the identity matrix and  $\mathbf{0}$  be the vector containing only zeros, both having appropriate dimensions according to their context. The Frobenius norm is denoted by  $\|\cdot\|_F$ . Let  $\mathbf{T}_{ij} \in \mathbb{R}^{d \times d}$  be an invertible transformation matrix aligning point-cloud  $\mathbf{X}_i$  with  $\mathbf{X}_j$  (for all  $i, j = 1, \dots, k$ ), where  $\mathbf{T}_{ij} = \mathbf{T}_{ji}^{-1}$ . Furthermore,  $\mathcal{T} = \{\mathbf{T}_{ij}\}_{i=1, j=1}^k$  is the set of all  $k^2$  pairwise transformations.

A desirable property of the set of transformations  $\mathcal{T}$  is that it complies with the following transitive consistency condition:

**Definition 1.** *The set of relative transformations  $\mathcal{T}$  is said to be transitively consistent if*

$$\mathbf{T}_{ij}\mathbf{T}_{jl} = \mathbf{T}_{il} \quad \text{for all } i, j, l = 1, \dots, k.$$

Definition 1 states that the transformation from  $i$  to  $j$  followed by the transformation from  $j$  to  $l$  must be the same as directly transforming from  $i$  to  $l$ .

**Lemma 1.** *The set of relative transformations  $\mathcal{T}$  is transitively consistent if and only if there is a set of invertible transformations  $\{\bar{\mathbf{T}}_i\}_{i=1}^k$  such that*

$$\mathbf{T}_{ij} = \bar{\mathbf{T}}_i \bar{\mathbf{T}}_j^{-1} \quad \text{for all } i, j = 1, \dots, k.$$

*Proof.* For the sake of completeness a proof is provided. “ $\Leftarrow$ ”: Transitive consistency of  $\mathcal{T}$  follows directly from the definition  $\mathbf{T}_{ij} = \bar{\mathbf{T}}_i \bar{\mathbf{T}}_j^{-1}$ , since for all  $i, j, l = 1, \dots, k$  it holds that

$$\mathbf{T}_{il} = \bar{\mathbf{T}}_i \bar{\mathbf{T}}_l^{-1} = \bar{\mathbf{T}}_i \bar{\mathbf{T}}_l^{-1} \tag{1}$$

$$= \bar{\mathbf{T}}_i (\bar{\mathbf{T}}_j^{-1} \bar{\mathbf{T}}_j) \bar{\mathbf{T}}_l^{-1} \tag{2}$$

$$= (\bar{\mathbf{T}}_i \bar{\mathbf{T}}_j^{-1}) (\bar{\mathbf{T}}_j \bar{\mathbf{T}}_l^{-1}) \tag{3}$$

$$= \mathbf{T}_{ij} \mathbf{T}_{jl}. \tag{4}$$

“ $\Rightarrow$ ”: In the rest of the proof we direct our attention towards the necessity of the existence of the  $\bar{\mathbf{T}}_i$  transformations.

First of all, if the transformations in  $\mathcal{T}$  are transitively consistent

$$\mathbf{T}_{ii} = \mathbf{I} \quad \text{for all } i = 1, \dots, k. \tag{5}$$

This follows by the fact that the  $\mathbf{T}_{ii}$  needs to be invertible while satisfying, by Definition 1, that  $\mathbf{T}_{ii}\mathbf{T}_{ii} = \mathbf{T}_{ii}$ .

Let  $\bar{\mathbf{T}}'_i = \mathbf{T}_{i1}$  for all  $i$ . Now we show that  $\bar{\mathbf{T}}'_i$  are such  $\bar{\mathbf{T}}_i$  matrices we seek. Since, by using (5),  $\bar{\mathbf{T}}'_1 = \mathbf{I}$ , we can write  $\mathbf{T}_{i1} = \bar{\mathbf{T}}'_i \mathbf{I} = \bar{\mathbf{T}}'_i (\bar{\mathbf{T}}'_1)^{-1}$ .

Now for any  $\mathbf{T}_{ij}$ , we can use that

$$\mathbf{T}_{1i}\mathbf{T}_{ij} = \mathbf{T}_{1j}. \tag{6}$$

Thus,

$$\mathbf{T}_{ij} = \mathbf{T}_{1i}^{-1}\mathbf{T}_{1j} = \mathbf{T}_{i1}\mathbf{T}_{j1}^{-1} = \bar{\mathbf{T}}'_i (\bar{\mathbf{T}}'_j)^{-1}. \tag{7}$$

□

### 2.2. Perfect Information

Due to Lemma 1, there is a reference coordinate frame, denoted by  $\star$ , from which there are  $\mathbf{T}_{i\star}$  transformations such that  $\mathbf{T}_{ij} = \mathbf{T}_{i\star}\mathbf{T}_{\star j}$  for all  $i, j$ . Note that the reference

coordinate frame is merely used as a tool for deriving our method and it is irrelevant what the actual reference frame is. Let us introduce

$$\mathbf{W} = [\mathbf{T}_{ij}] = \begin{bmatrix} \mathbf{T}_{11} & \cdots & \mathbf{T}_{1k} \\ \vdots & \ddots & \vdots \\ \mathbf{T}_{k1} & \cdots & \mathbf{T}_{kk} \end{bmatrix} \quad (8)$$

$$= \begin{bmatrix} \mathbf{T}_{1*} \mathbf{T}_{*1} & \cdots & \mathbf{T}_{1*} \mathbf{T}_{*k} \\ \vdots & \ddots & \vdots \\ \mathbf{T}_{k*} \mathbf{T}_{*1} & \cdots & \mathbf{T}_{k*} \mathbf{T}_{*k} \end{bmatrix} \quad (9)$$

$$= \begin{bmatrix} \mathbf{T}_{1*} \mathbf{T}_{1*}^{-1} & \cdots & \mathbf{T}_{1*} \mathbf{T}_{k*}^{-1} \\ \vdots & \ddots & \vdots \\ \mathbf{T}_{k*} \mathbf{T}_{1*}^{-1} & \cdots & \mathbf{T}_{k*} \mathbf{T}_{k*}^{-1} \end{bmatrix} \quad (10)$$

$$= \mathbf{U}_1 \mathbf{U}_2, \quad (11)$$

where

$$\mathbf{U}_1 = \begin{bmatrix} \mathbf{T}_{1*} \\ \mathbf{T}_{2*} \\ \vdots \\ \mathbf{T}_{k*} \end{bmatrix} \quad \text{and} \quad \mathbf{U}_2 = [\mathbf{T}_{1*}^{-1}, \mathbf{T}_{2*}^{-1}, \dots, \mathbf{T}_{k*}^{-1}].$$

Using this notation, finding either  $\mathbf{U}_1$  or  $\mathbf{U}_2$  (up to an invertible linear transformation) gives the transitively consistent transformations.

**Definition 2.** Let  $\mathbf{A} \in \mathbb{R}^{p \times q}$ . The set

$$\text{im}(\mathbf{A}) = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{R}^q\}$$

is the column space of  $\mathbf{A}$  and the set

$$\text{ker}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^q \mid \mathbf{Ax} = \mathbf{0}\}$$

is the null space of  $\mathbf{A}$ .

Note that due to the invertability of  $\mathbf{T}_{i*}$  (for all  $i = 1, \dots, k$ ) it holds that the matrix  $\mathbf{U}_1$  has rank  $d$  and thus the dimensionality of the column space  $\text{im}(\mathbf{U}_1)$  of  $\mathbf{U}_1$  is exactly  $d$ .

**Proposition 1.** Let  $\mathbf{Z} = \mathbf{W} - k\mathbf{I}$ . The linear subspace  $\text{ker}(\mathbf{Z})$  has dimension  $d$  and is equal to  $\text{im}(\mathbf{U}_1)$ .

*Proof.* First it is shown that the columns of  $\mathbf{U}_1$  are contained in the null space of  $\mathbf{Z}$ , i.e.  $\text{im}(\mathbf{U}_1) \subseteq \text{ker}(\mathbf{Z})$ , and then it is shown that the null space of  $\mathbf{Z}$  has exactly dimension  $d$ .

Note that  $\mathbf{U}_2 \mathbf{U}_1 = k\mathbf{I}$ , which we will make use of shortly. Multiplication of  $\mathbf{U}_1$  on the right to  $\mathbf{W} = \mathbf{U}_1 \mathbf{U}_2$  gives

$$\mathbf{WU}_1 = \mathbf{U}_1 \mathbf{U}_2 \mathbf{U}_1 = \mathbf{U}_1 k\mathbf{I} \quad (12)$$

$$\Leftrightarrow \mathbf{WU}_1 = k\mathbf{U}_1 \quad (13)$$

$$\Leftrightarrow \mathbf{WU}_1 - k\mathbf{U}_1 = \mathbf{0} \quad (14)$$

$$\Leftrightarrow (\mathbf{W} - k\mathbf{I})\mathbf{U}_1 = \mathbf{0} \quad (15)$$

$$\Leftrightarrow \mathbf{ZU}_1 = \mathbf{0} \quad \text{with } \mathbf{Z} = \mathbf{W} - k\mathbf{I}. \quad (16)$$

From (16) it can be seen that all the columns of  $\mathbf{U}_1$  are contained in the null space of  $\mathbf{Z}$ , so  $\text{im}(\mathbf{U}_1) \subseteq \text{ker}(\mathbf{Z})$ .

However, it still remains to be shown that the dimensionality of  $\text{ker}(\mathbf{Z})$  is exactly  $d$ , i.e.  $\text{im}(\mathbf{U}_1)$  spans the entire null space of  $\mathbf{Z}$  and not just a part of it. This is done by showing that there are no non-zero vectors  $\mathbf{x}$  that are not contained in  $\text{im}(\mathbf{U}_1)$  but are contained in  $\text{ker}(\mathbf{Z})$ .

Formally this is expressed by the requirement that the set

$$A = \{\mathbf{x} \in \mathbb{R}^{kd} \mid \mathbf{x} \neq \mathbf{0}, \mathbf{x} \notin \text{im}(\mathbf{U}_1), \mathbf{x} \in \text{ker}(\mathbf{Z})\}$$

is empty. Suppose now that  $A$  is not empty, so it contains the element  $\mathbf{x} \in \mathbb{R}^{kd}$ . Using the orthogonal decomposition theorem, the vector  $\mathbf{x}$  can be rewritten as  $\mathbf{x} = \mathbf{x}_{\text{ker}} + \mathbf{x}_{\text{im}}$ , where  $\mathbf{x}_{\text{ker}} \in \text{ker}(\mathbf{U}_1^T)$  and  $\mathbf{x}_{\text{im}} \in \text{im}(\mathbf{U}_1)$ . The definition of  $A$  states that  $\mathbf{x} \notin \text{im}(\mathbf{U}_1)$ , which implies that  $\mathbf{x}_{\text{ker}} \neq \mathbf{0}$ . Further, the definition of  $A$  states that

$$\mathbf{x} \in \text{ker}(\mathbf{Z}) \quad (17)$$

$$\Leftrightarrow \mathbf{Zx} = \mathbf{0} \quad (18)$$

$$\Leftrightarrow \mathbf{Z}(\mathbf{x}_{\text{ker}} + \mathbf{x}_{\text{im}}) = \mathbf{0} \quad (19)$$

$$\Leftrightarrow \mathbf{Zx}_{\text{ker}} + \mathbf{Zx}_{\text{im}} = \mathbf{0}. \quad (20)$$

Per definition  $\mathbf{x}_{\text{im}} \in \text{im}(\mathbf{U}_1) \subseteq \text{ker}(\mathbf{Z})$ , so it follows that  $\mathbf{Zx}_{\text{im}} = \mathbf{0}$ , leading to

$$\mathbf{Zx}_{\text{ker}} = \mathbf{0}. \quad (21)$$

Multiplication of  $\mathbf{x}_{\text{ker}}^T$  on the left gives

$$\mathbf{x}_{\text{ker}}^T \mathbf{Zx}_{\text{ker}} = 0 \quad (22)$$

$$\Leftrightarrow \mathbf{x}_{\text{ker}}^T (\mathbf{U}_1 \mathbf{U}_2 - k\mathbf{I})\mathbf{x}_{\text{ker}} = 0 \quad (23)$$

$$\Leftrightarrow \mathbf{x}_{\text{ker}}^T \mathbf{U}_1 \mathbf{U}_2 \mathbf{x}_{\text{ker}} - k\mathbf{x}_{\text{ker}}^T \mathbf{x}_{\text{ker}} = 0 \quad (24)$$

$$\Leftrightarrow \mathbf{x}_{\text{ker}}^T \mathbf{U}_2^T \mathbf{U}_1^T \mathbf{x}_{\text{ker}} - k\mathbf{x}_{\text{ker}}^T \mathbf{x}_{\text{ker}} = 0. \quad (25)$$

Per definition  $\mathbf{x}_{\text{ker}} \in \text{ker}(\mathbf{U}_1^T)$ , so  $\mathbf{U}_1^T \mathbf{x}_{\text{ker}} = \mathbf{0}$ , leading to

$$\mathbf{x}_{\text{ker}}^T \mathbf{x}_{\text{ker}} = 0 \quad (26)$$

$$\Leftrightarrow \mathbf{x}_{\text{ker}} = \mathbf{0}. \quad (27)$$

Equation (27) is a contradiction to  $\mathbf{x}_{\text{ker}} \neq \mathbf{0}$ , thus, the set  $A$  is empty.  $\square$

Proposition 1 states that  $\mathbf{U}_1$  in (16) can, up to an invertible linear transformation, be retrieved by finding the  $d$ -dimensional null space of  $\mathbf{Z}$ . Let  $\mathbf{Z} = \mathbf{U}\Sigma\mathbf{V}^T$  be the singular value decomposition (SVD) of  $\mathbf{Z}$ . The  $d$  columns of  $\mathbf{V}$  corresponding to the zero singular values span  $\text{ker}(\mathbf{Z})$  and give a solution to (16).

As we are only able to retrieve the transformations  $\mathbf{T}_{i*}$  in the blocks of  $\mathbf{U}_1$  up to invertible linear transformations,

w.l.o.g. we create a new version of  $\mathbf{U}_1$ , call it  $\mathbf{U}'_1$ , with the first  $d \times d$  block being equal to the identity, as

$$\mathbf{U}'_1 = \mathbf{U}_1 \mathbf{T}_{1\star}^{-1} = \begin{bmatrix} \mathbf{T}_{1\star} \mathbf{T}_{1\star}^{-1} \\ \mathbf{T}_{2\star} \mathbf{T}_{1\star}^{-1} \\ \vdots \\ \mathbf{T}_{k\star} \mathbf{T}_{1\star}^{-1} \end{bmatrix}. \quad (28)$$

### 2.3. Noisy Pairwise Transformations

Up until this point, the matrix  $\mathbf{U}_1$  is obtained under perfect information, *i.e.* the transitivity condition in Definition 1 holds for all  $\mathbf{T}_{ij}$  transformations contained in the blocks of  $\mathbf{W}$ . However, we are interested in the case when the transitivity condition does not hold due to measurement noise. Assume now that we have a noisy observation of  $\mathbf{W}$ , denoted as  $\tilde{\mathbf{W}}$ . Also, let the noisy version of  $\mathbf{Z}$  be  $\tilde{\mathbf{Z}} = \tilde{\mathbf{W}} - k\mathbf{I}$ . Now, in general it is not the case that the null space of  $\tilde{\mathbf{Z}}$  is  $d$ -dimensional. Instead, the least-squares approximation of the  $d$ -dimensional null space is considered, which leads to the following optimisation problem:

**Problem 1.** *Least-squares Transformation Synchronisation*

$$\begin{aligned} & \underset{\hat{\mathbf{T}}_{1\star}, \dots, \hat{\mathbf{T}}_{k\star}}{\text{minimise}} && \|\tilde{\mathbf{Z}} \hat{\mathbf{U}}_1\|_F^2 \\ & \text{subject to} && \mathbf{u}_i^T \mathbf{u}_j = 0 \quad \text{for all } i \neq j \\ & && \|\mathbf{u}_i\| = 1 \quad \text{for all } i \\ & && \hat{\mathbf{U}}_1 = [\mathbf{u}_1, \dots, \mathbf{u}_d] \in \mathbb{R}^{kd \times d}. \end{aligned}$$

The rank- $d$  approximation of the null space of  $\tilde{\mathbf{Z}}$  can be retrieved using the SVD of  $\tilde{\mathbf{Z}} = \mathbf{U}\Sigma\mathbf{V}^T$ . In this case the columns of  $\mathbf{V}$  corresponding to the  $d$  smallest singular values span the rank- $d$  approximation of the null space of  $\tilde{\mathbf{Z}}$ , giving  $\hat{\mathbf{U}}_1$ , the estimate for  $\mathbf{U}_1$ . By using (28),  $\hat{\mathbf{U}}'_1$  can be retrieved from  $\hat{\mathbf{U}}_1$ .

### 2.4. Affine Transformations in Homogeneous Coordinates

In this section it is shown that the method is also applicable for invertible affine transformations, rather than invertible linear transformations. This is done by representing the  $d$ -dimensional affine transformations  $\mathbf{T}_{ij}$  by using  $(d+1) \times (d+1)$  homogeneous matrices.

Each affine transformation  $\mathbf{T}_{ij}$  can be written as

$$\mathbf{T}_{ij} = \begin{bmatrix} \mathbf{A}_{ij} & \mathbf{0} \\ \mathbf{t}_{ij} & 1 \end{bmatrix}, \quad (29)$$

where  $\mathbf{A}_{ij}$  is the (invertible) linear  $d \times d$  transformation matrix and  $\mathbf{t}_{ij}$  is the  $d$ -dimensional row vector representing the translation. The inverse of  $\mathbf{T}_{ij}$  is given by

$$\mathbf{T}_{ij}^{-1} = \begin{bmatrix} \mathbf{A}_{ij}^{-1} & \mathbf{0} \\ -\mathbf{t}_{ij} \mathbf{A}_{ij}^{-1} & 1 \end{bmatrix}. \quad (30)$$

Similar to the linear case described in (8), the matrix  $\mathbf{W}$  is constructed from all  $\mathbf{T}_{ij}$ . It is assumed that the matrix  $\tilde{\mathbf{W}}$ , corresponding to the noisy observation of  $\mathbf{W}$ , contains blocks that are proper affine transformations, *i.e.* the last column of each block is  $[\mathbf{0} \ 1]^T$ .

A simple way to ensure that the synchronised transformations are affine transformations in homogeneous coordinates is to add the row vector  $\mathbf{z} = [z \ z \ \dots \ z] \in \mathbb{R}^{k(d+1)}$ , with  $z = [0 \ 0 \ \dots \ 0 \ 1] \in \mathbb{R}^{d+1}$ , to the matrix  $\tilde{\mathbf{Z}}$ . By adding the vector  $\mathbf{z}$  to  $\tilde{\mathbf{Z}}$ , the vector  $\mathbf{z}^T$  is removed from the null space of  $\tilde{\mathbf{Z}}$ . Using this approach, a solution is then found by solving Problem (1) with the updated  $\tilde{\mathbf{Z}}$ . Then the resulting  $\hat{\mathbf{U}}'_1$  gives an estimate of the first  $d$  columns of  $\hat{\mathbf{T}}_{i\star}$  ( $i = 1, \dots, k$ ) and these are the columns we seek.

### 2.5. Similarity Transformations

Similarity transformations are transformations that allow for translations, isotropic scaling, rotations and reflections. To retrieve similarity transformations, the estimates of the synchronised affine transformations  $\hat{\mathbf{T}}_{i\star}$  ( $i = 1, \dots, k$ ) are determined first. The translation component  $\hat{\mathbf{t}}_{i\star}$  of  $\hat{\mathbf{T}}_{i\star}$  can directly be extracted from  $\hat{\mathbf{T}}_{i\star}$  since it has the structure presented in (29). To obtain the scaling factor and the orthogonal transformation, the linear component  $\hat{\mathbf{A}}_{i\star}$  is factorised using SVD, resulting in  $\hat{\mathbf{A}}_{i\star} = \mathbf{U}_{i\star} \Sigma_{i\star} \mathbf{V}_{i\star}^T$ . The orthogonal component  $\hat{\mathbf{Q}}_{i\star}$  is then given by

$$\hat{\mathbf{Q}}_{i\star} = \mathbf{U}_{i\star} \mathbf{V}_{i\star}^T, \quad (31)$$

and the isotropic scaling factor  $\hat{s}_{i\star}$  is given by

$$\hat{s}_{i\star} = \left( \prod_{j=1}^d |(\sigma_{i\star})_{jj}| \right)^{\frac{1}{d}}, \quad (32)$$

where  $(\sigma_{i\star})_{jj}$  is the  $j$ -th element on the diagonal of  $\Sigma_{i\star}$ .

**Remark 1.** *It can be shown that retrieving the orthogonal component as presented in eq. (31) is the least-squares solution to the projection onto the set of orthogonal matrices. However, in eq. (32) the isotropic scaling factor is retrieved as the geometric mean of the individual axis-aligned scaling factors. The least-squares solution to the projection onto the set of similarity transformations is given by the arithmetic mean, *i.e.*  $\hat{s}_{i\star}^{lsq} = \frac{1}{d} \sum_{j=1}^d |(\sigma_{i\star})_{jj}|$ .*

### 2.6. Euclidean Transformations

Similarity transformations without isotropic scaling are called euclidean transformations. To obtain euclidean transformations, the similarity transformations are extracted and the scaling factors  $\hat{s}_{i\star}$  (for all  $i = 1, \dots, k$ ) are set to 1.

## 2.7. Rigid Transformations

Euclidean transformations without reflections are called rigid transformations. Rigid transformations can be obtained by ensuring that the determinant of the rotational component  $\hat{\mathbf{Q}}_{i\star}$  described in (31) equals 1. This can be achieved by setting

$$\hat{\mathbf{Q}}_{i\star} = \mathbf{U}_{i\star} \mathbf{D}_{i\star} \mathbf{V}_{i\star}^T, \text{ with} \quad (33)$$

$$\mathbf{D}_{i\star} = \text{diag}(1, \dots, 1, \det(\mathbf{V}_{i\star}^T \mathbf{U}_{i\star})). \quad (34)$$

## 3. Experiments

By generating ground truth data and adding Gaussian noise to it, we first compare the error of the synchronised transformations using our method to the error of the unsynchronised transformations. Furthermore, the transformation synchronisation method is applied for solving the Generalised Procrustes Problem with missing points and with wrong correspondence assignments.

### 3.1. Noisy Transformations

In this section it is described how the ground truth transformations are generated, how noisy versions thereof are generated and eventually results of the transformation synchronisation method are presented.

#### 3.1.1 Ground Truth Transformations

For the analysis of the performance of our method we generate a set of random transformations  $\mathcal{T}_\star = \{\mathbf{T}_{i\star}\}_{i=1}^k$ , that are used in turn to generate the transitively consistent set of pairwise transformations  $\mathcal{T} = \{\mathbf{T}_{ij} = \mathbf{T}_{i\star} \mathbf{T}_{j\star}^{-1}\}_{i=1, j=1}^k$ , serving as ground truth for the evaluation. The generation of  $\mathcal{T}_\star$  is described in the following.

The dot-notation is used to illustrate that  $\dot{x}$  is a random variable with a particular probability distribution. For generating the set  $\mathcal{T}_\star$ , we assume that the point-clouds that lead to the transformations have some structural similarity, *i.e.* the transformations are not entirely random. In particular, the scaling factors, the translation components and the linear part of the transformation are restricted in the sense that they cannot be arbitrary. However, arbitrary orientations in  $d$ -dimensional space are allowed for.

The set  $\mathcal{T}_\star$  contains the elements  $\mathbf{T}_{i\star}$  ( $i = 1, \dots, k$ ), which are samples of

$$\dot{\mathbf{T}} = \begin{bmatrix} \dot{s} \dot{\mathbf{Q}} \dot{\mathbf{N}} & \mathbf{0} \\ \dot{\mathbf{t}} & 1 \end{bmatrix}, \quad (35)$$

where  $\dot{s} \sim \mathcal{U}(0.5, 1.5)$  is a scaling factor and  $\dot{\mathbf{t}} \sim \mathcal{U}(-2.5, 2.5)^d$  is a translation, with  $\mathcal{U}(a, b)^d$  denoting the  $d$ -dimensional uniform distribution having the open interval  $(a, b)^d$  as support. Samples of the  $d \times d$  random rotation matrix  $\dot{\mathbf{Q}}$  are drawn by extracting the

rotational component of a non-singular random matrix as described in (31). The  $d \times d$  random noise matrix  $\dot{\mathbf{N}}$  is given by  $\dot{\mathbf{N}} = \mathbf{I} + \dot{\epsilon}$ , where  $\dot{\epsilon} \sim \mathcal{N}(0, 0.1^2)^{d \times d}$  is a  $d \times d$  random matrix with each element having univariate normal distribution  $\mathcal{N}(0, 0.1^2)$ . The purpose of creating the noise in the way using the random matrix  $\dot{\mathbf{N}}$  is to restrict the linear component in the transformation and thus to avoid ill-conditionedness with very high probability.

Depending on the type of transformation that is evaluated, the parameters of  $\mathcal{T}_\star$  have different properties, which are summarised in Table 1.

	$\dot{\mathbf{Q}}$	$\dot{\mathbf{t}}$	$\dot{s}$	$\dot{\mathbf{N}}$
linear	$ \det  = 1$	$= \mathbf{0}$	$\sim \mathcal{U}(0.5, 1.5)$	$\sim \mathbf{I} + \epsilon$
affine	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$\sim \mathcal{U}(0.5, 1.5)$	$\sim \mathbf{I} + \epsilon$
similarity	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$\sim \mathcal{U}(0.5, 1.5)$	$= \mathbf{I}$
euclidean	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$= 1$	$= \mathbf{I}$
rigid	$\det = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$= 1$	$= \mathbf{I}$

Table 1. Properties of components of random transformations for different types of transformations generated according to (35).

Once the ground truth set  $\mathcal{T}$  of transitively consistent transformations has been established, a noisy version thereof is synthetically created, as described in the next section.

The error  $e(\mathcal{T}^1, \mathcal{T}^2)$  between two sets of pairwise transformations  $\mathcal{T}^1 = \{\mathbf{T}_{ij}^1\}_{i,j=1}^k$  and  $\mathcal{T}^2 = \{\mathbf{T}_{ij}^2\}_{i,j=1}^k$  is defined as

$$e(\mathcal{T}^1, \mathcal{T}^2) = \frac{1}{k^2} \sum_{i,j=1}^k \|\mathbf{T}_{ij}^1 - \mathbf{T}_{ij}^2\|_F. \quad (36)$$

#### 3.1.2 Additive Gaussian Noise

The set of noisy pairwise transformations  $\tilde{\mathcal{T}}^{\mathcal{N}}$  is created by adding to each element of the matrix  $\mathbf{T}_{ij}$  a sample from  $\mathcal{N}(0, \sigma^2)$ , which is conducted for all matrices  $\mathbf{T}_{ij} \in \mathcal{T}$  with  $i \neq j$ . In the case of homogeneous transformation matrices  $\mathbf{T}_{ij}$ , no noise is added to the last column, which shall always be  $(\mathbf{0} \ 1)^T$ .

Results of the simulations are shown in Fig. 1. The first row of graphs show that for all types of transformations the error of synchronised transformations is smaller than the error of the unsynchronised transformations and that the slope of the error in the synchronised case is smaller than in the unsynchronised case. In the second row it can be seen that, even with a high amount of noise ( $\sigma = 0.5$ ), the error of the synchronised transformations decreases with an increasing number of objects  $k$ . As anticipated, with increasing  $k$  there is more information available, directly resulting in a lower error. The last row of graphs shows that increasing the dimensionality results in an increasing error; however, the error of the synchronised transformations increases slower than for the unsynchronised ones.

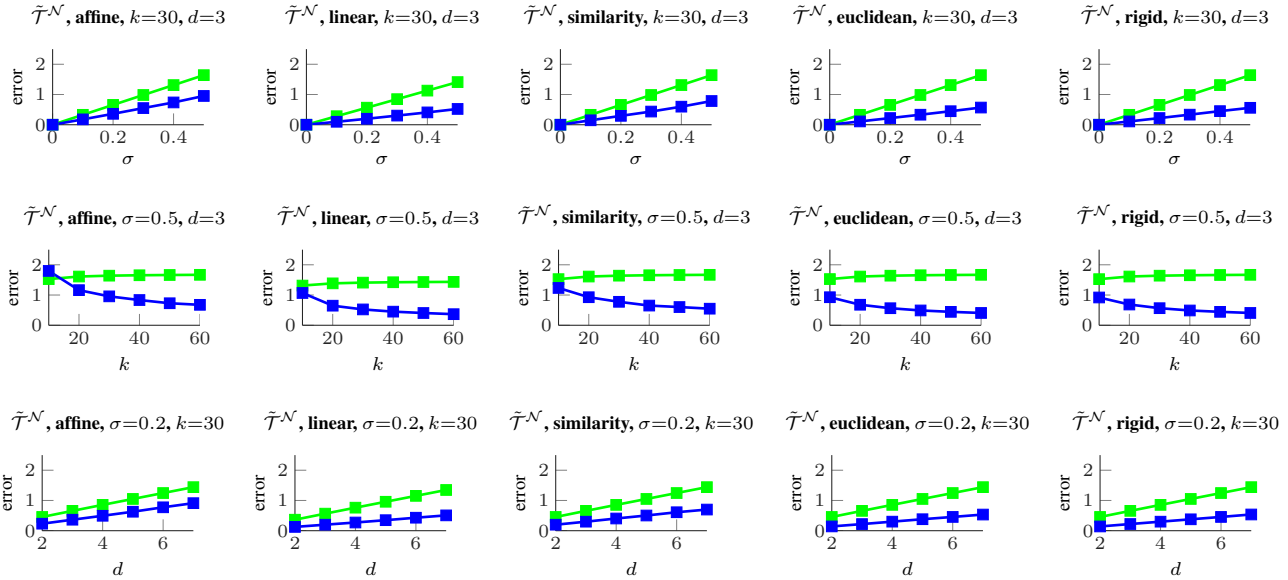


Figure 1. Error for additive normal noise  $\tilde{\mathcal{T}}^{\mathcal{N}}$  for different configurations as specified in the graph title with one varying parameter (horizontal axis). Each row of graphs shows a particular varying parameter ( $\sigma$ ,  $k$  and  $d$  from top to bottom) and each column of graphs shows a particular transformation type (affine, linear, similarity, euclidean, rigid, from left to right). The error as defined in (36) of the unsynchronised noisy transformations is shown in green and of the synchronised transformations in blue. Shown is the average error of 100 randomly generated sets of ground truth transformations, where for each ground truth transformation 20 runs of adding noise have been performed, resulting in a total of 2000 simulations per graph.

### 3.2. Generalised Procrustes Analysis

In addition to evaluating the synchronisation of noisy pairwise transformations we have applied our method for solving the Generalised Procrustes Problem (GPP), which is done on the one hand with missing data and on the other hand with wrong correspondence assignments. For both simulations the 2D fish shapes from the Chui-Rangarajan data set [4] with different levels of deformation and noise have been used (refer [4] for more details). For each level of deformation and noise the data set contains  $K = 100$  shapes, each comprising  $n = 98$  points in  $d = 2$  dimensions.

Finding the similarity transformation that best aligns two shapes, which is a subroutine for the evaluated reference-based, the iterative mean shape-based and the synchronisation-based method, is performed by an AOP implementation with symmetric scaling factors [13]. In the reference-based solution of GPP one shape is randomly selected as reference and all other shapes are aligned with the reference. For the iterative mean shape-based method the initial reference shape is selected randomly and then the mean shape is iteratively updated. In the synchronisation-based solution of GPP all  $k^2$  pairwise AOPs are solved first, followed by the synchronisation of the resulting transformations in order to aggregate all information contained in the pairwise transformations. Additionally, the stratified GPA

method proposed in [2] is evaluated for solving the GPP. In our experiments we have observed that by using the stratified GPA method the linear part of the resulting transformations may collapse to the zero matrix; in order to enable a comparison with the other methods in these cases the linear part of the transformation has simply been set to the identity matrix.

In the missing data experiments as well as the wrong correspondence experiments for each single run  $k = 30$  out of  $K = 100$  shapes are randomly selected. For the experiments in the missing points case the missing points are simulated by discarding points according to a given probability. For the experiments with wrong correspondences the correct correspondences are randomly disturbed in order to simulate wrong correspondences.

In contrast to solving the AOPs, in both experiments the computation of the error is performed using the original shape (*i.e.* with all points and with perfect correspondences). With that we investigate up to which amount recovering the original shapes from corrupt shape data is possible. The average shape error of a set of shapes  $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^k$  is defined as  $e(\mathcal{X}) = \frac{1}{k^2} \sum_{i,j=1}^k \|\mathbf{X}_i - \mathbf{X}_j\|_F$ .

#### 3.2.1 Missing Points

In every run, additionally to randomly selecting 30 out of 100 shapes, each data point of a shape is considered to

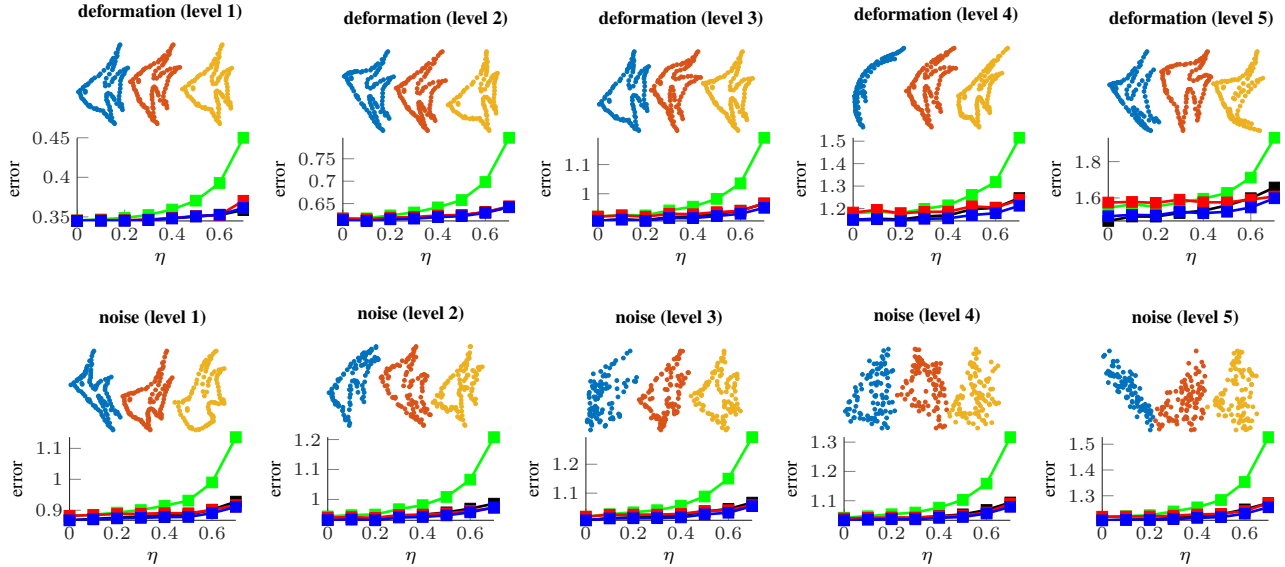


Figure 2. Average shape error for the reference-based (green), iterative mean shape-based (black), synchronisation-based (blue) and stratified method for solving the GPP with missing data. The horizontal axis shows the probability  $\eta$  that a point is considered missing. At the top of each graph three shapes according to the particular level of deformation or noise are depicted. Shown is the average shape error for 500 draws of missing data in each graph. In every run  $k = 30$  out of  $K = 100$  shapes are randomly selected, where each shape comprises  $n = 98$  points in  $d = 2$  dimensions.

be missing with probability  $\eta$ . As the implemented methods solve the AOP only for common points in each pair of shapes, values for  $\eta$  larger than 0.7 have not been investigated because with  $\eta > 0.7$  the cases that the number of common points in a pair of shapes is less than  $d = 2$  occur too frequently (for  $d$ -dimensional data, there must be at least  $d$  points in each shape in order to result in a system that is not under-determined). Also, for  $\eta \leq 0.7$  it is possible that the number of common points in a pair of shapes is less than  $d = 2$ ; in these cases the draw of missing data is simply repeated.

In Fig. 2 the resulting error of the reference-based, the iterative mean shape-based, the synchronisation-based and the stratified solution of the GPP with missing data are shown for different levels of deformation and noise. It can be seen that even with an increasing amount of missing data, when using the synchronisation-based method the error increases only slightly, whilst the error of the reference-based method increases significantly with a larger amount of missing points. With respect to the error, the transformation synchronisation method performs only marginally better than the iterative mean shape-based method and the stratified method. However, the average runtimes for solving a single GPP instance was 0.007 s for the reference-based method, 0.162 s for the synchronisation-based method, 1.932 s for the iterative mean shape-based method and 2.265 s for the stratified method, illustrating that our method performs significantly better than all other methods when taking runtime

and error into account at the same time.

### 3.2.2 Wrong Correspondence Assignments

Additionally to the case of missing points, we have applied our method to solve the GPP with wrong correspondence assignments between shapes. In order to mimic practical applications, where it is frequently the case that the true correspondences are unknown and thus it must be assumed that wrong correspondences are present, we do not make any efforts to correct these wrong correspondences (such as using RANSAC [7] or permutation synchronisation [14]). Instead, for each pair of shapes the AOP is solved whilst being aware that some of the points in the one shape have wrong counterparts in the other shape. Of course this will have influence on the resulting transformations. Thus, the objective of the simulations described in this section is to assess to what extent the transformations from shapes with wrong correspondences can be reconstructed using transformation synchronisation.

In every run, additionally to randomly selecting 30 out of 100 shapes, the correspondences between the  $n$  points in each shape are disturbed. For disturbing the correspondence assignments each pair of shapes that is to be aligned is considered independently. For that, a proportion of  $\nu \in [0, 1]$  points from the total number of  $n$  points is selected. Then, as correspondences between the pair of point-clouds  $\mathbf{X}_i, \mathbf{X}_j \in \mathbb{R}^{n \times 2}$  are implicitly given by the ordering of the rows, the rows corresponding to the previ-

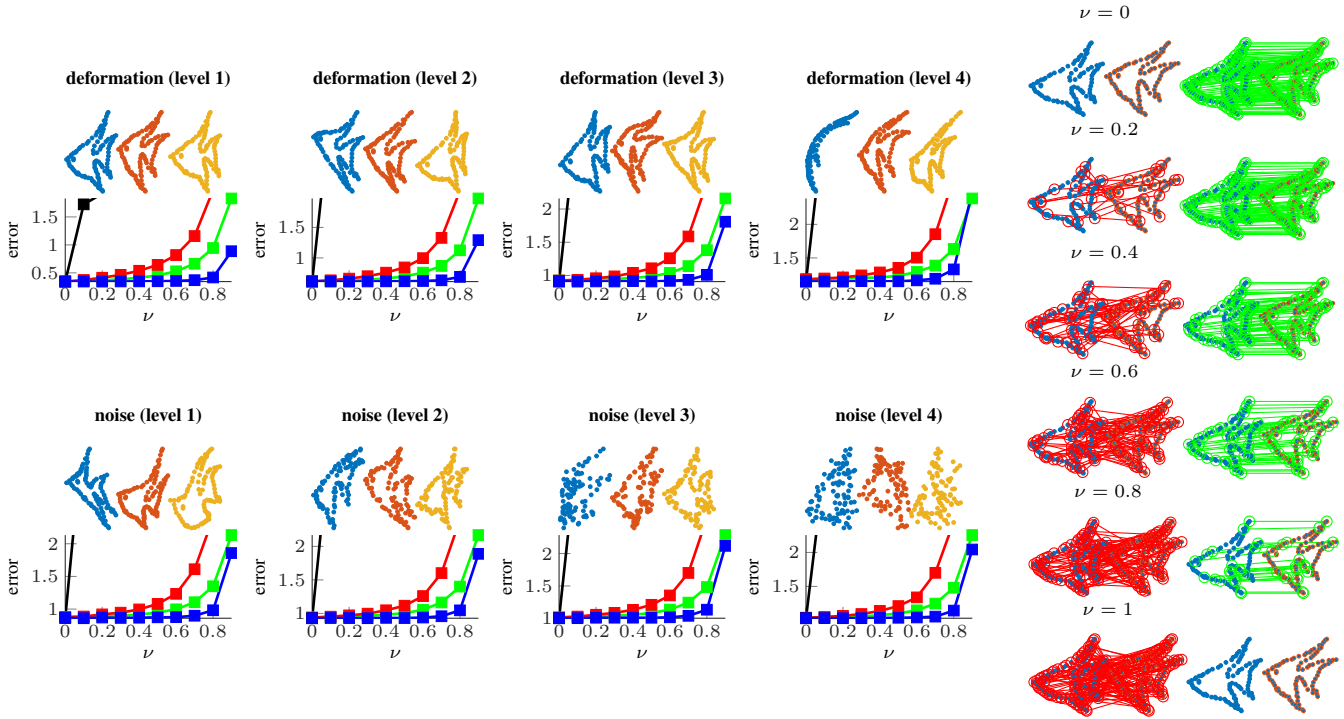


Figure 3. Average shape error for the reference-based (green), iterative mean shape-based (black), synchronisation-based (blue) and stratified (red) method for solving the GPP with wrong correspondences. The horizontal axis shows the proportion  $\nu$  of wrong correspondences. At the top of each graph three shapes according to the particular level of deformation or noise are depicted. Shown is the average shape error for 500 runs of disturbing correspondence assignments in each graph. In every run  $k = 30$  out of  $K = 100$  shapes are randomly selected, where each shape comprises  $n = 98$  points in  $d = 2$  dimensions. In the right-most column examples of the correspondence assignments between a pair of shapes are depicted for different values of  $\nu$  in each row. In order to keep the visualisation as coherent as possible, the wrong correspondences (red lines) and the correct correspondences (green lines) are shown separately.

ously selected points are reordered randomly in one of the point-clouds, directly resulting in disturbed correspondence assignments between the pair of point-clouds  $\mathbf{X}_i, \mathbf{X}_j$ .

In Fig. 3 the reference-based, the iterative mean shape-based, the synchronisation-based and the stratified solution of the GPP with wrong correspondences are shown for different levels of deformation and noise. On the right of Fig. 3 examples of the correspondences between pairs of shapes are depicted for different values of  $\nu$ .

It can be seen that for different levels of deformation and different levels of noise with 70% – 80% of wrong correspondences the outcome is only marginally affected when using our proposed method. In contrast, all other evaluated methods result in significantly larger errors, which can be explained by the fact that our method is the only one that is able to make use of the information that is contained in all pairwise transformations.

#### 4. Conclusion

The alignment of multiple (corresponding) point-clouds simultaneously is generally tackled by iteratively aligning all point-clouds to some reference. Whereas this approach

is biased (selecting a fixed reference) or initialisation-dependent (using the adaptive mean as reference) we have presented a method that is completely unbiased and does not depend on initialisation.

Our key observation is that the underlying noise-free transformations can be retrieved from the null space of a matrix that can directly be obtained from pairwise alignments. Whilst related approaches for rotation matrices [9, 10, 17] or permutation matrices [14] have been proposed, we have generalised the synchronisation method to handle general linear and affine transformations as well as similarity, euclidean and rigid transformations. Experimentally we were able to demonstrate that the proposed method is able to effectively reduce noise from the set of pairwise transformations and to solve the Generalised Procrustes Problem at least as good as existing approaches for the missing data case whilst significantly outperforming other methods for the presented wrong correspondence case.



## Acknowledgements

Supported by the Fonds National de la Recherche, Luxembourg (5748689, 6538106, 8864515).

## References

- [1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):698–700, 1987.
- [2] A. Bartoli, D. Pizarro, and M. Loog. Stratified generalized procrustes analysis. *International Journal of Computer Vision*, 101(2):227–253, 2013.
- [3] K. N. Chaudhury, Y. Khoo, and A. Singer. Global registration of multiple point clouds using semidefinite programming. *arXiv.org*, June 2013.
- [4] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.
- [5] T. F. Cootes and C. J. Taylor. Active Shape Models - Smart Snakes. In *In British Machine Vision Conference*, pages 266–275. Springer-Verlag, 1992.
- [6] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, Mar. 1997.
- [7] 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*, 24(6), June 1981.
- [8] J. C. Gower and G. B. Dijksterhuis. *Procrustes problems*, volume 3. Oxford University Press Oxford, 2004.
- [9] R. Hadani and A. Singer. Representation theoretic patterns in three dimensional Cryo-Electron Microscopy I: The intrinsic reconstitution algorithm. *Annals of mathematics*, 174(2):1219, 2011.
- [10] R. Hadani and A. Singer. Representation Theoretic Patterns in Three-Dimensional Cryo-Electron Microscopy II—The Class Averaging Problem. *Foundations of computational mathematics (New York, N.Y.)*, 11(5):589–616–616, 2011.
- [11] T. Heimann and H.-P. Meinzer. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, 2009.
- [12] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [13] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127, 1988.
- [14] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.
- [15] D. Pizarro and A. Bartoli. Global optimization for optimal generalized procrustes analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2409–2415. IEEE, 2011.
- [16] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, Mar. 1966.
- [17] A. Singer and Y. Shkolnisky. Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming. *SIAM journal on imaging sciences*, 4(2):543–572, June 2011.
- [18] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen Or. A survey on shape correspondence. In *Computer Graphics Forum*, pages 1681–1707. Wiley Online Library, 2011.
- [19] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, Nov. 1991.