Stratified Generalized Procrustes Analysis

Adrien Bartoli[1] **Daniel Pizarro**[2] **Marco Loog**[3]

[1] Clermont Université, France

[2] Universidad de Alcalá, Spain

[3] Delft University of Technology, The Netherlands

Corresponding author: Adrien Bartoli

`Adrien.Bartoli@gmail.com`

Abstract

Generalized procrustes analysis computes the best set of transformations that relate matched shape data. In shape analysis the transformations are usually chosen as similarities, while in general statistical data analysis other types of transformation groups such as the affine group may be used. Generalized procrustes analysis has a nonlinear and nonconvex formulation. The classical approach alternates the computation of a so-called reference shape and the computation of transformations relating this reference shape to each shape datum in turn.

We propose the stratified approach to generalized procrustes analysis. It first uses the affine transformation group to analyze the data and then upgrades the solution to the sought after group, whether Euclidean or similarity. We derive a convex formulation for each of these two steps, and efficient practical algorithms that gracefully handle missing data (incomplete shapes.)

Extensive experimental results show that our approaches perform well on simulated and real data. In particular our closed-form solution gives very accurate results for generalized procrustes analysis of Euclidean data.

Keywords: shape, registration, procrustes, theseus, generalized.

Implementation: our generalized procrustes analysis toolbox which implements the methods we propose and methods from the literature in Matlab is freely available under the GPL licence. Download the code from `http://isit.u-clermont1.fr/~ab/SGPAv1.0.tgz`.

Contents

1 Introduction

In many different problems, data analysis requires one to first compensate for a global transformation between the different datasets or *shape data*. This is known as *procrustes analysis* in the statistics and shape analysis literature (Dryden and Mardia, 1998; Gower and Dijksterhuis, 2004). Each shape datum is a set of points. This analysis is called *generalized procrustes analysis* when more than two shape data are to be registered. In this problem, one global transformation per observed shape has to be computed, so that the shapes are mapped to a common coordinate frame whereby they look as 'similar' as possible. This process is called also *rigid registration*. For example, it is typical in landmark-based 2D and 3D shape analysis to compensate for a similarity (rotation, translation and scale factor) so as to normalize the shapes to be further analyzed. The global transformation is usually modeled by one of the groups of Euclidean transformations ($\frac{1}{2}d(d+1)$ degrees of freedom), similarity transformations ($\frac{1}{2}d(d+1)+1$ degrees of freedom) and affine transformations ($d(d+1)$ degrees of freedom), with d the dimension of the shape data to be analyzed. The classical approach to generalized procrustes analysis is to select one of the shapes as a *reference shape*, and register each of the other shapes to the reference in turn. It is common to then alternate a re-estimation of the reference shape, as the average of the registered shapes, with shape registration. We call this general paradigm the *alternation approach* to generalized procrustes analysis.[1] The paradigm of estimating both the registrations and a reference shape is statistically grounded, and we shall use it in this paper.

We propose the *stratified approach* to generalized procrustes analysis. We initially published it in a shorter conference paper (Bartoli et al., 2010). Our stratified generalized procrustes analysis approach draws on recent results and practice from the 3D reconstruction or Structure-from-Motion community (Hartley and Zisserman, 2003), whereby one estimates a so-called uncalibrated camera model and then upgrades or self-calibrates it. Here, the uncalibrated model is the affine transformation group, and the calibrated model is one of the similarity or Euclidean transformation groups. Our stratified approach has several advantages: it processes data in batch, as opposed to the alternation approach, and gracefully deals with missing data (incomplete shapes.) Our experimental results show that our methods perform well on challenging datasets.

This paper is organized as follows. We state the problem and give mathematical preliminaries in §2. We review previous work in §3. Our stratified approach is outlined in §4, and its two main components, namely affine generalized procrustes analysis and similarity-Euclidean generalized procrustes analysis are respectively given in §§5 and 6. We report experimental results on simulated and real data on various datasets in §7. We finally conclude in §8. The next two paragraphs summarize our contributions and give

[1] The *alternation* term comes from the Structure-from-Motion literature where there exist methods that alternate the estimation of the camera motion with the estimation of the scene structure, see (Mahamud et al., 2001) for instance.

our notation.

Contributions. The contributions we bring in this paper can be summarized as follow. First, we propose an alternative formulation of generalized procrustes analysis: the data-space model, as opposed to the classical reference-space model. Second, we propose a closed-form solution to affine generalized procrustes analysis with missing data in the reference-space model. Third, we propose a method to upgrade or 'convert' the affine result into an initial similarity or Euclidean solution that can then be refined by means of nonlinear least squares. This affine initialization followed by upgrading is what we call the stratified approach to generalized procrustes analysis.

Notation. We write matrices in sans-serif fonts (*e.g.*, S) and vectors in bold (*e.g.*, $\mathbf{S}_j$.) Scalars are in italics lower case and transformations in italics upper case (see below.) Matrix inverse is written as in A^{-1}, transpose as in $\mathsf{A}^\top$ (also for vectors as in $\mathbf{S}_j^\top$), pseudo-inverse as in $\mathsf{S}^\dagger$ and the hat matrix as $\hat{\mathsf{S}} \overset{\text{def}}{=} \mathsf{S}\mathsf{S}^\dagger$. We use an SVD (Singular Value Decomposition) based pseudo-inverse, applicable to non-square and column-rank-deficient matrices.[2] For a full-column-rank portrait matrix S, the pseudo-inverse is also given by $\mathsf{S}^\dagger = (\mathsf{S}^\top\mathsf{S})^{-1}\mathsf{S}^\top$. The operator diag constructs a diagonal or a block-diagonal matrix from a vector or a list of matrices, respectively. The group of orthonormal matrices $O(d)$ in dimension d is defined as $\mathsf{Q} \in O(d) \Leftrightarrow (\mathsf{Q} \in \mathbb{R}^{d\times d}, \quad \mathsf{Q}^\top\mathsf{Q} = \mathrm{I})$ where I is the identity matrix of appropriate size. The special group of orthonormal matrices $SO(d)$ is used to represent rotations in dimension d and is defined as $\mathsf{R} \in SO(d) \Leftrightarrow (\mathsf{R} \in O(d), \quad \det(\mathsf{R}) = 1)$. We use nnz to indicate the number of non-zero elements in a matrix. For $\mathbf{u} \in \mathbb{R}^d$, we define the following two operators:

$$\nu(\mathbf{u}) \overset{\text{def}}{=} \begin{pmatrix} \mathbf{u}^\top & & \\ & \ddots & \\ & & \mathbf{u}^\top \end{pmatrix} \in \mathbb{R}^{d\times d^2}.$$

We write vect the row-wise matrix vectorization operator. The 'all-one' and 'all-zero' vectors are written $\mathbf{1}$ and $\mathbf{0}$. We define $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top\mathbf{u}}$ as the vector two-norm of $\mathbf{u}$ and $\|\mathsf{U}\|_{\mathcal{F}} \overset{\text{def}}{=} \sqrt{\mathrm{tr}(\mathsf{U}^\top\mathsf{U})}$ as the matrix Frobenius norm of U.

[2] $\mathsf{S}^\dagger$ is the Moore-Penrose pseudo-inverse of S, that in practice is obtained using an SVD:

$$\mathsf{S} \overset{\mathsf{SVD}}{\to} \mathsf{U}\Sigma\mathsf{V}^\top \quad \text{and} \quad \mathsf{S}^\dagger \overset{\text{def}}{=} \mathsf{V}\Sigma^\dagger\mathsf{U}^\top,$$

where the entries of the diagonal matrix $\mathbf{\Sigma}$ (assumed in decreasing order) are pseudo-inversed as $\sigma^\dagger = 0$ if $\sigma < \epsilon$ and $\frac{1}{\sigma}$ otherwise, with ϵ a small constant.

2 Problem Statement and Mathematical Preliminaries

We now give a mathematical statement of generalized procrustes analysis that will later be used to derive our approach and to describe previous work. We present both the data-space and the reference-space models. In both cases, one has to estimate an unknown reference shape $\mathsf{S}^\top \overset{\text{def}}{=} \begin{pmatrix} \mathbf{S}_1 & \cdots & \mathbf{S}_m \end{pmatrix} \in \mathbb{R}^{d\times m}$, with $\mathbf{S}_j \in \mathbb{R}^d$ a shape point, and n unknown global transformations $\mathcal{T} \overset{\text{def}}{=} \{T_1, \ldots, T_n\}$, $T_i : \mathbb{R}^d \to \mathbb{R}^d$, $i = 1, \ldots, n$, given n shape data $\mathsf{D}_i^\top \overset{\text{def}}{=} \begin{pmatrix} \mathbf{D}_{i,1} & \cdots & \mathbf{D}_{i,m} \end{pmatrix} \in \mathbb{R}^{d\times m}$, with $\mathbf{D}_{i,j} \in \mathbb{R}^d$ a shape datum point. So as to model *missing data*, the fact that some points may not be observed in some shape data, we define the binary visibility variables $v_{i,j} \in \{0, 1\}$ with $v_{i,j} = 0$ for a missing point and $v_{i,j} = 1$, otherwise. We define $\alpha \overset{\text{def}}{=} \sum_{i=1}^n \sum_{j=1}^m v_{i,j}$ the total number of observed shape points, $\beta_i \overset{\text{def}}{=} \sum_{j=1}^m v_{i,j}$ the number of shape points observed in shape i and $\gamma_j \overset{\text{def}}{=} \sum_{i=1}^n v_{i,j}$ the number of observations of shape point number j. Note that $\sum_{i=1}^n \beta_i = \sum_{j=1}^m \gamma_j = \alpha$ and that $\beta_i \leq m$, $\gamma_j \leq n$ and $\alpha \leq mn$.

While most previous work on standard and generalized procrustes analysis uses the reference-space model, as will become clear in §3, our stratified methods will use both models to try and find convex numerical solutions. The main difference between the two models lies in the cost function being minimized, and thus the assumption made on the noise distribution: the data-space model assumes the noise to be *i.i.d.* in the measurements (and is in this respect a generative model), while the reference-space model assumes the noise to be *i.i.d.* in the transformed shape space. In terms of practical computation, the two models are equivalent for Euclidean generalized procrustes analysis, and have different advantages and drawbacks.

2.1 The Reference-Space Model

The *reference-space model* is illustrated by figure 1. In this model, one minimizes the discrepancy between the unknown reference shape S and the registered shape data $T_i^{-1}(\mathsf{D}_i)$ for $i = 1, \ldots, n$, under a set of problem dependent constraints $\tilde{\mathcal{C}}$ to avoid that the solution be degenerate. This model thus tries to maximize the agreement between the shape data after they are aligned in a common coordinate frame, here the reference coordinate frame. The minimization problem is stated as:

$$\arg\min_{\mathcal{T},\mathsf{S}} \tilde{\mathcal{R}}(\mathcal{T}, \mathsf{S}) \qquad \text{s.t.} \qquad \tilde{\mathcal{C}}(\mathcal{T}, \mathsf{S}) \;=\; \mathsf{0}.$$

Assuming all shape points available in all shapes, then the reference-space cost is given by:

$$\tilde{\mathcal{R}}(\mathcal{T}, \mathsf{S}) \;=\; \sum_{i=1}^{n} \|T_i^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2 \tag{1}$$

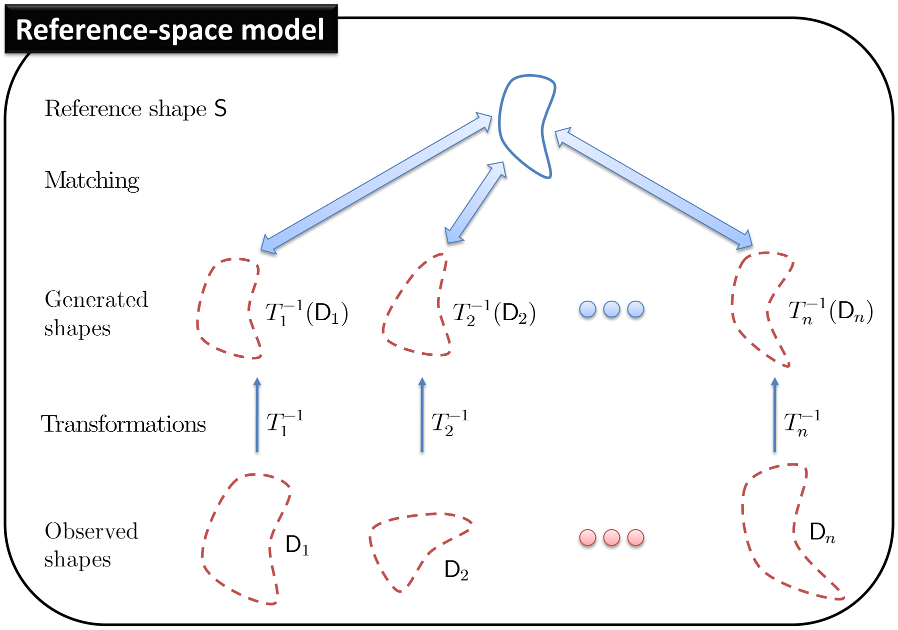


Figure 1: **The reference-space model.** The shape data are compared to the reference shape in the coordinate system of the reference shape, that is to say, after registration. Minimizing the reference-space cost is optimal in the sense of maximum likelihood if the residuals are Gaussian *i.i.d.* for the reference shape points.

This is the cost minimized in standard generalized procrustes analysis. Missing data are simply modeled by expanding cost (1) over the shape point index $j = 1, \ldots, m$ and introducing the binary visibility variables $v_{i,j}$ as:

$$\tilde{\mathcal{R}}(\mathcal{T}, \mathsf{S}) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j} \, \|T_i^{-1}(\mathbf{D}_{i,j}) - \mathbf{S}_j\|_2^2. \tag{2}$$

2.2 The Data-Space Model

The *data-space model* is illustrated by figure 2. It is related to a generative modeling of the data. The reason for the name ‘data-space’ shall become clear shortly. The motivation for this model is related to the physics of imaging and sensing in general. A sensor can be theoretically modeled using an ideal process. In practice however, a sensor outputs measurements subject to noise. These measurements will generally be described as a combination of the theoretical sensor model and a model of noise. Being generative, our data-space model follows this idea that the sensor noise occurs in the observations. It thus consists in basing the error function on the discrepancy between the model-predicted shapes $T_i(\mathsf{S}) \in \mathbb{R}^{d \times m}$ and the observed shapes $\mathsf{D}_i \in \mathbb{R}^{d \times m}$ for $i = 1, \ldots, n$. The error function must be minimized under a set of constraints $\mathcal{C}$ ensuring that the solution be non-degenerate (an example of constraints is $\text{rank}(\mathsf{S}) - d = 0$, where the operator rank

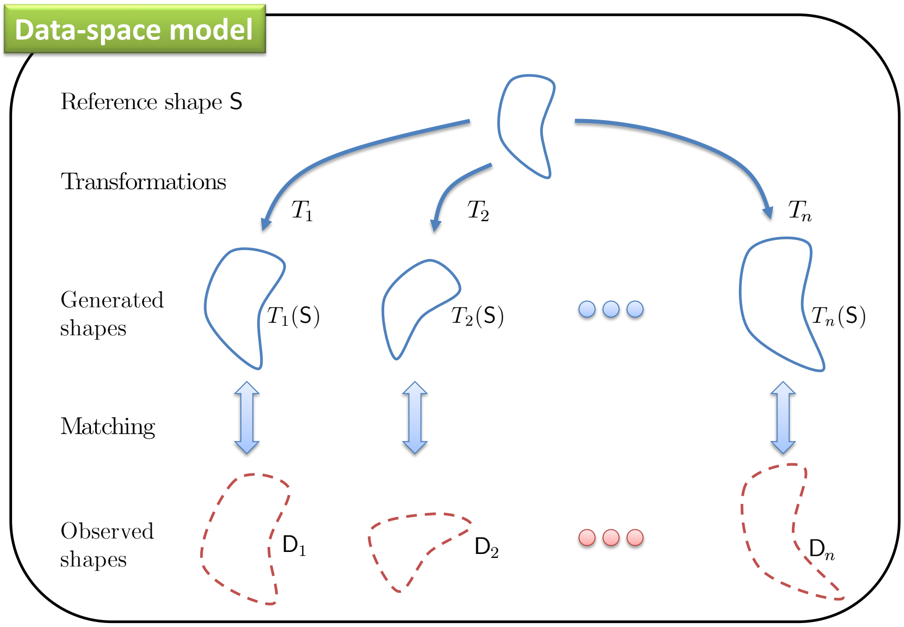


Figure 2: **The data-space model.** Minimizing the data-space cost is a generative approach. It is optimal in the sense of maximum likelihood if the residuals (the error on the observed shape points) are Gaussian *i.i.d.*

computes the rank of a matrix), as is discussed later in the paper. These constraints may also fix some of the *gauge freedoms*.[3] The number of gauge freedoms is the number of degrees of freedom of the global transformation group being considered. The minimization problem is stated as:

$$\arg\min_{\mathcal{T},\mathsf{S}} \mathcal{R}(\mathcal{T},\mathsf{S}) \qquad \text{s.t.} \qquad \mathcal{C}(\mathcal{T},\mathsf{S}) \;=\; 0.$$

Assuming all shape points available in all data shapes, then the data-space cost $\mathcal{R}$ is given by:

$$\mathcal{R}(\mathcal{T},\mathsf{S}) \;=\; \sum_{i=1}^{n} \|\mathsf{D}_i - T_i(\mathsf{S})\|_{\mathcal{F}}^2. \tag{3}$$

This equation reveals the bilinear structure of the problem. As in the reference-space model, missing data are simply modeled by expanding cost (3) over the shape point index $j = 1, \ldots, m$ and introducing the binary visibility variables $v_{i,j}$ as:

$$\mathcal{R}(\mathcal{T},\mathsf{S}) \;\overset{\text{def}}{=}\; \sum_{i=1}^{n}\sum_{j=1}^{m} v_{i,j}\;\|\mathbf{D}_{i,j} - T_i(\mathbf{S}_j)\|_2^2. \tag{4}$$

[3] The gauge is the coordinate frame used to represent the unknowns. It is changed by a *gauge transformation*, lying in the *gauge group*.

The data-space model is frequently used in model estimation from data, as for instance in the close problem of Structure-from-Motion. It gives the Maximum Likelihood Estimate if the noise on the observed data is *i.i.d.* Gaussian. It matches the definition of (Yezzi and Soatto, 2003) of registration based on the notion of average shape. The data-space cost is *gauge invariant*: it does not depend on the coordinate frame in which the reference shape $\mathsf{S} \in \mathbb{R}^{m \times d}$ and the transformations $\mathcal{T}$ are expressed. Let $G : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be any invertible transformation representing a gauge transformation: the set of transformations $\mathcal{T} \circ G^{-1} = \{T_1 \circ G^{-1}, \dots, T_n \circ G^{-1}\}$ and reference shape $G(\mathsf{S}) = \begin{pmatrix} G(\mathbf{S}_1) & \cdots & G(\mathbf{S}_m) \end{pmatrix}^\top$ leave the data-space cost unchanged, as is easily verified.

The data-space cost (4) is generally nonlinear and nonconvex. Consequently, there does not exist a general closed-form solution to the maximum likelihood problem for the data-space model. In the particular case of affine registration with no missing data (*i.e.*, all points observed in all shape data), there exist a solution method that deals with any dimension and number of shape data, since this particular problem can be cast as a rank-d matrix factorization problem. Even though there does not seem to exist a paper that would describe this method in details, it seemed common practice to us, especially in Structure-from-Motion. We also noted that some authors developed methods to handle missing data, for instance by filling-in the missing parts of the measurement matrix (Aguiar et al., 2008). For all the other cases, one has to resort to iterative nonlinear least squares optimization. This raises the problem of finding an initial solution lying as close as possible to the global minimum of the data-space cost (4). Our stratified approach provides such a 'good' initial solution.

2.3 More Modeling Details

We now give more modeling details, in particular on the ambiguities and the number of points needed for solving generalized procrustes analysis.

Shape reflection. The reflection effect happens when a shape datum is the reflection of another shape datum. In practice, it causes the rotational part of the transformation, whether affine, similarity or Euclidean, to have a negative determinant. At this point, we have to choose whether we want to allow or prevent the shape data to be reflected by the estimated transformations. One can enforce non-reflexion by forcing the determinant of some $(d \times d)$ matrices to be positive, which means order-d polynomial inequalities. We made the following choice: we let the affine transformations reflect the shape data while we force the similarity-Euclidean transformations to be non-reflecting. The reason is that the affine case may apply to general statistical data analysis, while the similarity-Euclidean case applies to shape data only. In our stratified approach we will of course ensure that the affine transformations are non-reflecting prior to upgrading them to similarity-Euclidean transformations.

Transformations. A Euclidean transformation $T_E = (\mathsf{E}; \mathbf{e})$ is represented by a matrix $\mathsf{E} \in SO(d)$ (the rotation) and a vector $\mathbf{e} \in \mathbb{R}^d$ (the translation.) It is applied to a point $\mathbf{S} \in \mathbb{R}^d$ as:

$$\mathbf{S} \quad \mapsto \quad T_E(\mathbf{S}) \quad \stackrel{\text{def}}{=} \quad \mathsf{E}\mathbf{S} + \mathbf{e}. \tag{5}$$

A similarity transformation $T_S = (\zeta; \mathsf{E}; \mathbf{e})$ has the same representation with a scalar $\zeta > 0$ (the scale factor; being positive it prevents reflections.) It is applied to a point $\mathbf{S} \in \mathbb{R}^d$ as:

$$\mathbf{S} \quad \mapsto \quad T_S(\mathbf{S}) \quad \stackrel{\text{def}}{=} \quad \zeta\mathsf{E}\mathbf{S} + \mathbf{e}. \tag{6}$$

Finally, an affine transformation $T_A = (\mathsf{A}; \mathbf{a})$ is represented by a matrix $\mathsf{A} \in \mathbb{R}^{d\times d}$ (the 'rotational' part) and a vector $\mathbf{a} \in \mathbb{R}^d$ (the translation.) It is applied to a point $\mathbf{S} \in \mathbb{R}^d$ as:

$$\mathbf{S} \quad \mapsto \quad T_A(\mathbf{S}) \quad \stackrel{\text{def}}{=} \quad \mathsf{A}\mathbf{S} + \mathbf{a}. \tag{7}$$

We note that the affine transformation can be written as:

$$T_A(\mathbf{S}) \quad = \quad \nu(\mathbf{S})\text{vect}(\mathsf{A}) + \mathbf{a}.$$

An affine transformation factors in a purely affine transformation and a Euclidean transformation using QR decomposition of its rotational part A:

$$\mathsf{A} \quad \rightarrow \quad \mathsf{Q}\mathsf{Z}, \tag{8}$$

where $\mathsf{Q} \in O(d)$ and $\mathsf{Z} \in \mathbb{R}^{d\times d}$ is an upper triangular matrix. If $\det(\mathsf{Q}) = -1$ switching the sign of the last column of Q and of $Z_{d,d}$ ensures $\mathsf{Q} \in SO(d)$, *i.e.*, that Q is a proper rotation matrix. The possible reflection is then contained in matrix Z. In this decomposition, Z represents the 'purely affine' part of the transformation and Q the Euclidean part, leading to:

$$T_A(\mathbf{S}) \quad = \quad \mathsf{Q}\mathsf{Z}\mathbf{S} + \mathbf{a}. \tag{9}$$

We note that the purely affine transformations, whose rotational part is represented by an upper triangular matrix, form a group.

Abusing notation, we use a letter such as T for the generic $\mathbb{R}^d \rightarrow \mathbb{R}^d$ transformation function, and use the same letter for the transformation parameters over which a minimization should be conducted. Sets of transformations are written using calligraphic fonts such as $\mathcal{T} = \{T_1, \ldots, T_n\}$. We also directly apply a transformation to a whole shape $\mathsf{S} \in \mathbb{R}^{m \times d}$ as for instance $\mathsf{S} \rightarrow \mathsf{S}\mathsf{E}^\top + \mathbf{1}\mathbf{e}^\top$ in the Euclidean case.

Minimum number of data points. It is clear that generalized procrustes analysis requires a minimum number of $n \geq 2$ observed shapes. While in practice we would like to apply a registration framework to a redundant set of data points, it is worth deriving what the minimum number of data points permitting registration is, and if there exist degenerate geometric cases. Generally speaking, each data point provides d equations on the corresponding unknown shape. Let e be the number of degrees of freedom, or adjustable parameters, for the transformation group being considered (for instance, $e = d(d+1)$ in the affine case.) In the case of $n = 2$ shapes, we need at least $\frac{e}{d}$ observed points shared by the two shapes (it is obvious that a point seen in one shape but not in the other does no provide any equation.) Generalizing to an arbitrary number of shapes, we conclude that each shape must share at least $\frac{e}{d}$ observed points with another observed shape. Finally, we would like to underline that there exist cases for which the equations are dependent, and may lead to so-called degenerate configurations for which despite the number of points is theoretically high enough, there is more than a single solution to the registration problem. We give two simple examples of dependent constraints. The first one is straightforward. Three colinear points give $2d$ equations only, and not $3d$ equations, as can be easily verified from the above given transformation models. The second example is on the estimation of a single Euclidean transformation between two observed shapes with $d = 3$. A straightforward use of the formulae shows that $e = \frac{1}{2}d(d+1) = 6$, and that therefore, $\frac{e}{d} = \frac{6}{3} = 2$ shared points should be enough to estimate the Euclidean transformation. However, only 5 out of the 6 equations provided by 2 points are independent. This can easily be seen by considering the rotation taking as axis the line joining the two points (for the shape taken as target of the transformation.) A rotation around this axis with an arbitrary angle can be applied to the computed transformation, leaving one degree of freedom free. As a concluding remark, we recommend one to numerically check whereby a set of observed shape points form a degenerate configuration, by directly inspecting the numerical conditioning of the systems of equations that have to be solved at runtime.

Since in our stratified approach shown in table 1 we start off with affine registration in all cases, we require the input data to always provide enough constraints for affine registration, which implies that a subsequent similarity-Euclidean registration is well-constrained too.[4]

[4] Note however that the approach could be used with a number of points sufficient for similarity-Euclidean registration but insufficient for affine registration. For that special case, one could compute a subspace of affine registrations, whose parameters would be fixed in a second step by the orthonormality constraints of the similarity-Euclidean registration.

2.4 On the Relationship Between the Reference-Space and the Data-Space Models

The reference-space and the data-space models both lead one to minimize a cost over the reference shape $\mathsf{S} \in \mathbb{R}^{m \times d}$ and the set of transformations T_i, $i = 1, \ldots, n$, respectively given by equations (2) and (4). Both costs look similar:

$$\begin{array}{ll} \text{Reference-space cost} & \text{Data-space cost} \\ \tilde{\mathcal{R}}(\tilde{\mathcal{T}}, \tilde{\mathsf{S}}) \;=\; \sum_{i=1}^{n} \|\tilde{T}_i^{-1}(\mathsf{D}_i) - \tilde{\mathsf{S}}\|_{\mathcal{F}}^2 & \mathcal{R}(\mathcal{T}, \mathsf{S}) \;=\; \sum_{i=1}^{n} \|\mathsf{D}_i - T_i(\mathsf{S})\|_{\mathcal{F}}^2 \end{array}$$

We here analyze what are exactly the differences between these two costs. In other words, we want to characterize the differences between the minimizers $\tilde{\mathcal{T}}^\star$ and $\tilde{\mathsf{S}}^\star$ of $\tilde{\mathcal{R}}$, and the minimizers $\mathcal{T}^\star$ and $\mathsf{S}^\star$ of $\mathcal{R}$. For that purpose, we shall distinguish amongst the three types of afore-mentioned transformations: Euclidean, similarity and affine.

Euclidean transformations. We instanciate the data-space cost with n Euclidean transformations $T_{E,i} = (\mathsf{E}_i; \mathbf{e}_i)$, $i = 1, \ldots, n$:

$$\mathcal{R}(\mathcal{T}_E, \mathsf{S}) \;=\; \sum_{i=1}^{n} \|\mathsf{D}_i - T_{E,i}(\mathsf{S})\|_{\mathcal{F}}^2.$$

Each term can be rewritten as:

$$\|\mathsf{D}_i - T_{E,i}(\mathsf{S})\|_{\mathcal{F}}^2 \;=\; \|\mathsf{D}_i^\top - \mathsf{E}_i \mathsf{S}^\top - \mathbf{e}_i \mathbf{1}^\top\|_{\mathcal{F}}^2 \;=\; \|\mathsf{E}_i^\top \mathsf{D}_i^\top - \mathsf{S}^\top - \mathsf{E}_i^\top \mathbf{e}_i \mathbf{1}^\top\|_{\mathcal{F}}^2.$$

The last equality follows from $\mathsf{E}_i \in SO(d)$. Because $T_{E,i}^{-1} = (\mathsf{E}_i^\top; -\mathsf{E}_i^\top \mathbf{e}_i)$, this leads to:

$$\|\mathsf{D}_i - T_{E,i}(\mathsf{S})\|_{\mathcal{F}}^2 \;=\; \|T_{E,i}^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2.$$

This equation implies that both costs are identical ($\tilde{\mathcal{R}}(\mathcal{T}_E, \mathsf{S}) = \mathcal{R}(\mathcal{T}_E, \mathsf{S})$). Therefore, both models share the same solution $\tilde{\mathcal{T}}^\star = \mathcal{T}^\star$ and $\tilde{\mathsf{S}}^\star = \mathsf{S}^\star$.

Similarity transformations. We follow the same steps as in the Euclidean transformations case. Let $T_{S,i} = (\zeta_i; \mathsf{E}_i; \mathbf{e}_i)$, we obtain for the model-space cost:

$$\mathcal{R}(\mathcal{T}_S, \mathsf{S}) \;=\; \sum_{i=1}^{n} \|\mathsf{D}_i - T_{S,i}(\mathsf{S})\|_{\mathcal{F}}^2.$$

Each term can be rewritten as:

$$\|\mathsf{D}_i - T_{S,i}(\mathsf{S})\|_{\mathcal{F}}^2 \;=\; \|\mathsf{D}_i^\top - \zeta_i \mathsf{E}_i \mathsf{S}^\top - \mathbf{e}_i \mathbf{1}^\top\|_{\mathcal{F}}^2 \;=\; \zeta_i^2 \|\frac{1}{\zeta_i} \mathsf{E}_i^\top \mathsf{D}_i^\top - \mathsf{S}^\top - \frac{1}{\zeta_i} \mathsf{E}_i^\top \mathbf{e}_i \mathbf{1}^\top\|_{\mathcal{F}}^2.$$

The last equality follows from $\mathsf{E}_i \in SO(d)$ and $\|\zeta \mathsf{A}\|_{\mathcal{F}}^2 = \zeta^2 \|\mathsf{A}\|_{\mathcal{F}}^2$. Because $T_{S,i}^{-1} = (\frac{1}{\zeta_i}; \frac{1}{\zeta_i}\mathsf{E}_i^\top; -\frac{1}{\zeta_i}\mathsf{E}_i^\top \mathbf{e}_i)$, this leads to:

$$\|\mathsf{D}_i - T_{S,i}(\mathsf{S})\|_{\mathcal{F}}^2 \quad = \quad \zeta_i^2 \|T_{S,i}^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2.$$

This equation implies that the two costs differ by the weight given to each shape data:

Reference-space cost	Data-space cost
$\tilde{\mathcal{R}}(\mathcal{T}_S, \mathsf{S}) \;=\; \sum_{i=1}^n \|\tilde{T}_{S,i}^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2$	$\mathcal{R}(\mathcal{T}_S, \mathsf{S}) \;=\; \sum_{i=1}^n \zeta_i \|\tilde{T}_{S,i}^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2$

The weighting is directly given by the scale ζ_i.

Affine transformations. We introduce a set of affine transformations $T_{A,i} = (\mathsf{A}_i; \mathbf{a}_i)$ in the model-space cost:

$$\mathcal{R}(\mathcal{T}_A, \mathsf{S}) \quad = \quad \sum_{i=1}^n \|\mathsf{D}_i - T_{A,i}(\mathsf{S})\|_{\mathcal{F}}^2.$$

Each term can be rewritten as:

$$\|\mathsf{D}_i - T_{A,i}(\mathsf{S})\|_{\mathcal{F}}^2 \quad = \quad \|\mathsf{D}_i^\top - \mathsf{A}_i \mathsf{S}^\top - \mathbf{a}_i \mathbf{1}^\top\|_{\mathcal{F}}^2.$$

By using decomposition (9), giving $\mathsf{A}_i = \mathsf{Q}_i \mathsf{Z}_i$ with $\mathsf{Q}_i \in SO(d)$, we obtain:

$$\|\mathsf{D}_i - T_{A,i}(\mathsf{S})\|_{\mathcal{F}}^2 \quad = \quad \|\mathsf{D}_i^\top - \mathsf{Q}_i \mathsf{Z}_i \mathsf{S}^\top - \mathbf{a}_i \mathbf{1}^\top\|_{\mathcal{F}}^2 \quad = \quad \|\mathsf{Q}_i^\top \mathsf{D}_i^\top - \mathsf{Z}_i \mathsf{S}^\top - \mathsf{Q}_i^\top \mathbf{a}_i \mathbf{1}^\top\|_{\mathcal{F}}^2.$$

Under this form, each term is similar to the corresponding reference-space term. The difference lies in the upper triangular matrix Z_i. It cannot be simply inverted as an orthonormal matrix since it would change the value of the term. Because it represents the purely affine part of the transformation, we can state that the more Euclidean the sought transformations, the closer the reference-space and data-space models. More formally, assuming $\mathsf{Z}_i \approx \mathrm{I}$:

$$\|\mathsf{D}_i - T_{A,i}(\mathsf{S})\|_{\mathcal{F}}^2 \quad \approx \quad \|\mathsf{Z}_i^{-1} \mathsf{Q}_i^\top \mathsf{D}_i^\top - \mathsf{S}^\top - \mathsf{Z}_i^{-1} \mathsf{Q}_i^\top \mathbf{a}_i \mathbf{1}^\top\|_{\mathcal{F}}^2 \quad = \quad \|\mathsf{A}_i^{-1} \mathsf{D}_i^\top - \mathsf{S}^\top - \mathsf{A}_i^{-1} \mathbf{a}_i \mathbf{1}^\top\|_{\mathcal{F}}^2.$$

Because $T_{A,i}^{-1} = (\mathsf{A}_i^{-1}; -\mathsf{A}_i^{-1} \mathbf{a}_i)$, this leads to:

$$\|\mathsf{D}_i - T_{A,i}(\mathsf{S})\|_{\mathcal{F}}^2 \quad \approx \quad \|T_{A,i}^{-1}(\mathsf{D}_i) - \mathsf{S}\|_{\mathcal{F}}^2,$$

and thus $\tilde{\mathcal{T}}^\star \approx \mathcal{T}^\star$ and $\tilde{\mathsf{S}}^\star \approx \mathsf{S}^\star$. In other words, a model can be approximated by the other one, with a quality of approximation related to the amount of affine skewness.

3 Previous Work

The state-of-the-art on procrustes analysis is quite rich. We give it an overview in the context of statistical shape analysis. Procrustes analysis finds the best procrustes transformations between sets of points. A procrustes transformation is defined as a similarity, though it was originally defined as a Euclidean (orthonormal) transformation (Schönemann, 1966). Procrustes analysis has originally been defined for two shapes only (Schönemann, 1966; Schönemann and Carroll, 1970), and was later extended to generalized procrustes analysis, that deals with multiple shapes (Gower, 1975; Ten Berge, 1977). The two-shape problem is also dubbed the *absolute orientation problem* (Horn, 1987). The algorithms in the literature are almost all based on the idea of alternating the estimation of the transformations and of the reference shape. The latter is initially taken as one of the shape data, and the transformations are estimated independently for each shape datum, leading to the *alternation framework*. The algorithm terminates if the change in the reference shape or its distance to the registered shape data is sufficiently small:

1. Initialize the reference shape $\mathsf{S} \leftarrow \mathsf{D}_{i^\star}$ as one of the shape data – the shape with the highest number of points is usually chosen ($i^\star = \arg\max_i \beta_i$)
2. For $i = 1, \ldots, n$, Estimate the similarity T_i from S to the i-th shape datum, Endfor
3. Re-estimate the reference shape: $\mathsf{S} \leftarrow \frac{1}{n} \sum_{i=1}^{n} T_i^{-1}(\mathsf{D}_i)$
4. If the termination criterion is not met, go to step 2

Our termination criterion is met if the norm of the difference between between two consecutive reference shapes is lower than 10^{-6}. It should be noted that the alternation approach minimizes the reference-space[5] cost (1). This general framework can be adapted to handle shape data where some points are missing, and so to minimize the reference-space cost with missing data (2). Step 1 cannot initialize the whole reference shape from a single shape datum. Since the registrations are still to be computed, it is not at this stage possible to combine them, and the most ‘complete’ shape datum is used instead. While the registration proceeds, missing points in the reference shape are filled in. It might not be possible to register some of the shape data in the early iterations while the reference shape is getting completed. The alternation framework being iterative, it is not guaranteed to converge to a global minimum of the reference-space cost and depends on the initial reference shape chosen by the algorithm.

[5] While at step 2 each transformation can be estimated by minimizing a cost in the data-space or in the reference-space without changing the solution (the two different costs just differ by a scale factor), it makes a difference at step 3 when re-estimating the reference shape. Averaging the transformed shape data minimizes the reference-space cost conditioned on the current similarities.

The implementations of this general framework mostly differ on how they compute the data to reference shape similarity transformations, by solving the absolute orientation problem. This computation heavily depends on how the rotational part of the similarity transformations is represented, and what error criterion is minimized. Iterative solutions were proposed in the 1950s and 1960s, as (Eggert et al., 1997) reports. Closed-form solutions were then proposed using quaternions (Horn, 1987) and orthonormal matrices (Horn et al., 1988; Umeyama, 1991). Dual quaternions were also used to represent both the rotational and translational parts (Walker et al., 1991). These algorithms provide closed-form solutions, which is a nice feature. However, they minimize error criteria which are algebraic, and may not behave well in the presence of noise in the shape data (Eggert et al., 1997). Since in practice the shape data are always corrupted by noise or, even worse, may not at all satisfy the similarity transformation model, it is crucial that the registration procedure handles noise and deviation from the estimated model. For that reason, the estimated closed-form solution is often refined by iteratively minimizing some nonlinear, statistically well-founded solution, such as the negative log-likelihood of the shape datum to be registered, conditioned on the reference shape. Various solutions to the absolute orientation problem with noise-contaminated data were proposed (Arun et al., 1987; Goryn and Hein, 1995; Kanatani, 1994; Matei and Meer, 1999; Ohta and Kanatani, 1998; Ramos and Verriest, 1997). An integrated solution to multiple shape registration based on the above framework was also proposed with a total least squares registration step (Krishnan et al., 2005; Wen et al., 2005). (Xiao et al., 2006) recently proposed to simultaneous perform generalized procrustes analysis while estimating a model for the deformable component between a set of shape data.

The stratified approach we propose is different from the alternation approaches where reference shape estimation and shape datum registration are alternated. Alternation approaches need an initial solution while our approach uses convex optimization and finds an initial solution in closed-form. Our approach draws on ideas borrowed from stratified 3D reconstruction in Structure-from-Motion. It gracefully deals with missing data and improves computational efficiency and accuracy.

4 The Stratified Approach

Our contributions in this paper are the stratified approach to generalized procrustes analysis and a set of models and tools that implements it. The stratified approach makes it possible to efficiently register multiple shapes, in any dimension and for the three aforementioned types of transformations (affine, similarity and Euclidean.) Our framework has four main steps. The first two steps perform affine registration (initialization and nonlinear refinement, respectively), and are needed in all cases, except registration without missing data that is dealt with matrix factorization. The last two steps perform similarity-Euclidean registration (initialization from the affine registration and nonlinear refinement, respectively.) Our stratified framework is outlined in table 1. The four main steps are described below:

1. **Affine registration: reference-space solution (§5.)** We give a closed-form solution that minimizes the *reference-space cost*.

2. **Affine registration: data-space refinement.** The data-space cost can be efficiently locally minimized from the reference-space solution using classical Orthogonal Distance Regression (Boggs et al., 1989), as implemented in the context of Bundle Adjustment in Structure-from-Motion (Slama, 1980; Triggs et al., 2000).

3. **Similarity-Euclidean registration: initial solution (§6.)** Given an affine registration of multiple shapes, we show how both the transformations and the reference shape can be 'upgraded' to one of the similarity-Euclidean registrations in closed-form. This entails one to resolve the purely affine part of the internal gauge freedoms, and to project the affine transformations to the set of rotation matrices.

4. **Similarity-Euclidean registration: iterative refinement.** The data-space cost can be minimized as in the affine case directly above adapted to the similarity-Euclidean registration cases.

OBJECTIVE

Affine, similarity or Euclidean generalized procrustes analysis (registration of multiple matched shapes represented by d-dimensional points.)

INPUTS

The inputs are n shapes with m points. Visible and missing points are indicated by $v_{i,j} \in \{0, 1\}$, where $i = 1, \dots, n$ is the shape index and $j = 1, \dots, m$ is the point index.

OUTPUTS

The outputs are the n sought transformations and the m points of the reference shape.

ALGORITHM

- *Affine registration*
  If the data are complete (all points observed in all shape data, $v_{i,j} = 1$ for $i = 1, \dots, n$ and $j = 1, \dots, m$)
    Factorization-based registration
  Else
    Initial registration with the reference-space model (§5)
    Nonlinear refinement with Orthogonal Distance Regression

- *Similarity-Euclidean registration*
    Check that the determinant of the rotational part of all the input affine transformations share the same sign
    Upgrading from affine registration (§6)
    Nonlinear refinement with Orthogonal Distance Regression

Table 1: **The proposed stratified registration approach**.

5 Affine Registration in the Reference-Space

We define $\mathcal{A} \overset{\text{def}}{=} \{A_1, \dots, A_n\}$ to be the set of unknown affine transformations. An affine transformation A_i is represented as a pair $A_i = (\mathsf{A}_i; \mathbf{a}_i) \in \mathbb{R}^{d \times d} \times \mathbb{R}^d$. We present a closed-form solution to this problem in the reference-space.

5.1 Problem Statement

We first introduce the inverse $B_i = (\mathsf{B}_i; \mathbf{b}_i) \overset{\text{def}}{=} (\mathsf{A}_i^{-1}; -\mathsf{A}_i^{-1}\mathbf{a}_i)$ of the sought affine transformations (and the set $\mathcal{B} \overset{\text{def}}{=} \{B_1 \dots, B_n\}$), and instanciate the reference-space cost $\tilde{\mathcal{R}}$ from equation (2) as:

$$\tilde{\mathcal{R}}(\mathcal{B}, \mathsf{S}) \overset{\text{def}}{=} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j} \ \|\mathsf{B}_i \mathbf{D}_{i,j} + \mathbf{b}_i - \mathbf{S}_j\|_2^2.$$

The major advantage of the reference-space cost is that it is a sum of squares linear in the adjustable parameters, and thus leads to a linear least squares optimization problem.

We now set the optimization problem to be solved as:

$$\min_{\mathcal{B}, \mathsf{S}} \tilde{\mathcal{R}}(\mathcal{B}, \mathsf{S}) \qquad \text{s.t.} \qquad \text{rank}(\mathsf{S}) \ = \ d. \tag{10}$$

The non-degeneracy constraint we choose here is rank$(\mathsf{S}) = d$. It is necessary to prevent the reference shape to collapse, *i.e.*, to avoid the trivial zero-cost solution $\mathsf{S} = \mathsf{0}$ and $\mathcal{A} = \{(\mathsf{0}; \mathbf{0}), \dots, (\mathsf{0}; \mathbf{0})\}$. By deriving the gauge properties of the reference-space model, we shall see shortly that while sensible and theoretically well-founded, this rank-based non-degeneracy constraint needs in practice to be 'strengthened' to allow one to solve for the reference-space model.

5.2 Proposed Algorithm

Deriving our algorithm takes several steps. We first investigate the gauge properties of the reference-space model. We then formulate the problem in matrix form, and finally derive our closed-form solution.

Gauge freedoms and non-degeneracy. While the data-space cost is gauge-invariant, the reference-space cost is not. Recall that for affine registration, the gauge group is the group of affine transformations. We will show that the reference-space cost is 'sub-gauge' invariant. In other words, it is invariant to a subgroup of the gauge group, namely, the Euclidean group, and that the cost function extrema are invariant to the similarity group. Figure 3 shows that changing the gauge with the gauge transformation G gives, for the data-space model:

$$\mathsf{S} \to G(\mathsf{S}) \qquad \text{and} \qquad \mathcal{A} \to \mathcal{A} \circ G^{-1},$$

and for the reference-space model:

$$\mathsf{S} \to G(\mathsf{S}) \qquad \text{and} \qquad \mathcal{B} \to G \circ \mathcal{B},$$

as can easily be verified from figure 3. Plugging this gauge transformation into the reference-space cost (2) also shows that it is invariant only to Euclidean transforms and preserves its extrema under similarity transforms. In other words, the purely affine part of the gauge group changes the value of the reference-space cost. As a consequence, it might make the reference shape to get shrunk to $\mathsf{S} = \mathsf{O}$ (all point coordinates equal 0) while solving the reference-space model, artificially making the cost to vanish. So as to prevent this shrinking to happen, one has to enforce a strong non-degeneracy constraint, fixing the purely affine part of the gauge group. The $\text{rank}(\mathsf{S}) = d$ constraint is however a weak constraint: it does not guarantee that the reference shape is not *near-degenerate*. For that reason, we propose to use a stronger non-degeneracy constraint. The idea is to force the reference shape to be as non-degenerate as possible, by enforcing its covariance matrix to be the identity matrix. The non-degeneracy constraint thus becomes $\mathsf{S}^\top \mathsf{S} = \mathrm{I}$. We note that it implies $\text{rank}(\mathsf{S}) = d$, and fixes the purely affine and scaling parts of the gauge group.

For pure convenience in our optimization algorithm, as shall become clear in the next section, we choose to fix the origin of the coordinate frame at the centroid of the reference shape using the constraint $\mathsf{S}\mathbf{1}_{(m\times 1)} = \mathbf{0}_{(d\times 1)}$. Problem (10) is rewritten with the gauge constraints as:

$$\min_{\mathcal{B},\mathsf{S}} \tilde{\mathcal{R}}(\mathcal{B},\mathsf{S}) \qquad \text{s.t.} \qquad \mathsf{S}^\top\mathsf{S} \;=\; \mathrm{I}_{(d\times d)} \qquad \text{and} \qquad \mathsf{S}^\top\mathbf{1}_{(m\times 1)} = \mathbf{0}_{(d\times 1)}.$$

Matrix formulation. We define the matrix $\tilde{\mathsf{X}} \in \mathbb{R}^{p\times d}$ with $p \stackrel{\text{def}}{=} (d+1)n + m$. This matrix contains the unknowns as:

$$\tilde{\mathsf{X}}^\top \;\stackrel{\text{def}}{=}\; \begin{pmatrix} \mathsf{B}^\top & \mathsf{S}^\top \end{pmatrix} \qquad \text{with} \qquad \mathsf{B}^\top \;\stackrel{\text{def}}{=}\; \begin{pmatrix} \mathsf{B}_1 & \mathbf{b}_1 & \cdots & \mathsf{B}_n & \mathbf{b}_n \end{pmatrix}. \tag{11}$$

We use the notation $\tilde{\mathcal{R}}(\tilde{\mathsf{X}}) \stackrel{\text{def}}{=} \tilde{\mathcal{R}}(\mathcal{B},\mathsf{S})$. We have to find $\tilde{\mathsf{X}}$ by solving the following constrained optimization problem, where matrix $\mathsf{K} \in \mathbb{R}^{\alpha\times p}$ will be derived shortly (recall that α is the total number of observed shape points):

$$\min_{\tilde{\mathsf{X}}\in\mathbb{R}^{p\times d}} \|\mathsf{K}\tilde{\mathsf{X}}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top\mathsf{S} \;=\; \mathrm{I}_{(d\times d)} \qquad \text{and} \qquad \mathsf{S}^\top\mathbf{1}_{(m\times 1)} = \mathbf{0}_{(d\times 1)}. \tag{12}$$

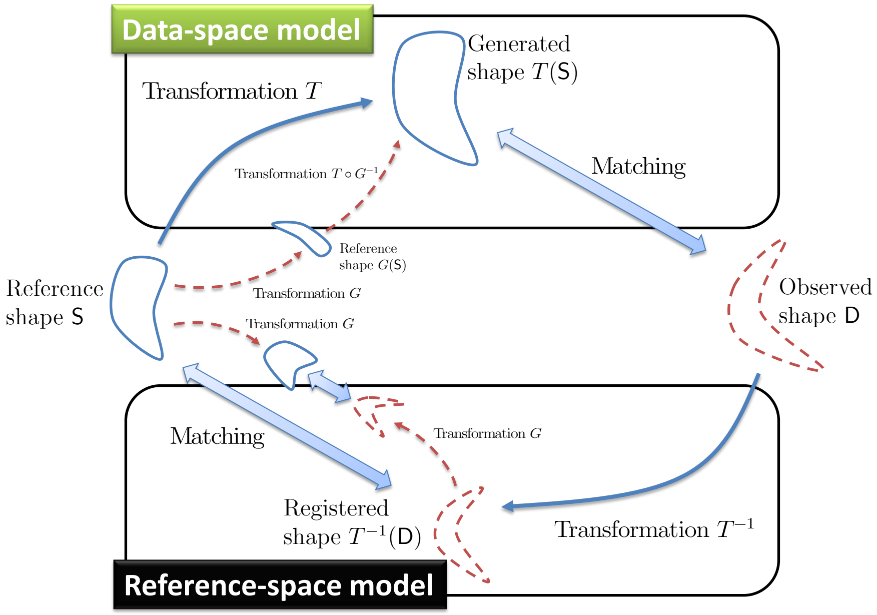

Data-space model
Transformation T
Generated shape $T(\mathsf{S})$
Matching
Transformation $T \circ G^{-1}$
Reference shape $G(\mathsf{S})$
Transformation G
Reference shape S
Transformation G
Observed shape D
Transformation G
Matching
Registered shape $T^{-1}(\mathsf{D})$
Transformation T^{-1}
Reference-space model

Figure 3: **Comparison of the gauge properties of the data-space and the reference-space models for affine generalized procrustes analysis.** The data-space cost is gauge invariant while the reference-space cost is sub-gauge-invariant, as explained in the main text.

An efficient minimization strategy is important so as to be able to cope with the potentially very large number of adjustable parameters. The problem at hand however is fortunatelly very well structured and 'sparse'. We first rewrite the reference-space cost (2) as:

$$\tilde{\mathcal{R}}(\tilde{\mathsf{X}}) = \sum_{i=1}^{n}\sum_{j=1}^{m} v_{i,j} \left\| \mathbf{D}_{i,j}^\top \mathsf{B}_i^\top + \mathbf{b}_i^\top - \mathbf{S}_j^\top \right\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} v_{i,j} \left\| \begin{pmatrix} \mathbf{D}_{i,j}^\top & 1 & -1 \end{pmatrix} \begin{pmatrix} \mathsf{B}_i^\top \\ \mathbf{b}_i^\top \\ \mathbf{S}_j^\top \end{pmatrix} \right\|_2^2 .$$

We organize the residuals by shape datum first, and rewrite $\tilde{\mathcal{R}}$ in matrix form. Matrix $\mathsf{K} \in \mathbb{R}^{\alpha \times p}$ has the following structure:

$$\mathsf{K} \overset{\text{def}}{=} \begin{pmatrix} \mathsf{K}_B & \mathsf{K}_S \end{pmatrix} \quad \text{with} \quad \mathsf{K}_B \overset{\text{def}}{=} \begin{pmatrix} \mathsf{K}_{B,1} & & \\ & \ddots & \\ & & \mathsf{K}_{B,n} \end{pmatrix} \in \mathbb{R}^{\alpha \times n(d+1)} \quad \text{and} \quad \mathsf{K}_S \overset{\text{def}}{=} \begin{pmatrix} \mathsf{K}_{S,1} \\ \vdots \\ \mathsf{K}_{S,n} \end{pmatrix} \in \mathbb{R}^{\alpha \times m},$$

with:

$$\mathsf{K}_{B,i} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{D}_{i,1} & 1 \\ \vdots & \vdots \\ \mathbf{D}_{i,m} & 1 \end{pmatrix} \in \mathbb{R}^{\beta_i \times d+1} \qquad \text{and} \qquad \mathsf{K}_{S,i} \overset{\text{def}}{=} - \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \in \mathbb{R}^{\beta_i \times m},$$

where only the 'rows' for which $v_{i,j} = 1$ are instanciated in $\mathsf{K}_{B,i}$ and $\mathsf{K}_{S,i}$.

It should be noted that matrix K is very sparse. Indeed, it is easy to see that the order of the number of non-zero entries in K is related to the total number of observed shape points α times the dimension d:

$$\text{nnz}(\mathsf{K}) = \sum_{i=1}^{n} (\text{nnz}(\mathsf{K}_{B,i}) + \text{nnz}(\mathsf{K}_{S,i})) = \sum_{i=1}^{n} (\beta_i(d+1) + \beta_i) = \alpha(d+2).$$

It might thus be possible to solve problem (12) using some direct sparse matrix factorization algorithm. This would however have two drawbacks. First, problem (12) involves a linear least squares cost, but is under nonlinear constraints. Handling these constraints breaks the efficiency of the sparse factorization algorithms. Second, the problem is highly structured, thanks to the shape of matrices K_B and K_S. A general purpose sparse matrix factorization algorithm would not take advantage of this block-structure. The closed-form solution we describe below gracefully handles the nonlinear constraints, and is designed to take advantage of the specific problem structure.

A closed-form solution. We rewrite problem (12) as:

$$\min_{\tilde{\mathsf{X}} \in \mathbb{R}^{p \times d}} \|\mathsf{K}_B \mathsf{B} + \mathsf{K}_S \mathsf{S}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top \mathsf{S} = \mathrm{I}_{(d \times d)} \qquad \text{and} \qquad \mathsf{S}^\top \mathbf{1}_{(m \times 1)} = \mathbf{0}_{(d \times 1)}. \tag{13}$$

Temporarily fixing S, we get the minimizer for B as:

$$\mathsf{B}^\star \overset{\text{def}}{=} -\mathsf{K}_B^\dagger \mathsf{K}_S \mathsf{S}. \tag{14}$$

Thanks to the structure of matrix K_B, the pseudo-inverse $\mathsf{K}_B^\dagger$ can be computed very efficiently, as shall be described shortly. Substituting the expression (14) for $\mathsf{B}^\star$ in the problem formulation (13), we get:

$$\min_{\mathsf{S} \in \mathbb{R}^{m \times d}} \| - \mathsf{K}_B \mathsf{K}_B^\dagger \mathsf{K}_S \mathsf{S} + \mathsf{K}_S \mathsf{S}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top \mathsf{S} = \mathrm{I}_{(d \times d)} \qquad \text{and} \qquad \mathsf{S}^\top \mathbf{1}_{(m \times 1)} = \mathbf{0}_{(d \times 1)}.$$

Matrix $\hat{\mathsf{K}}_B \overset{\text{def}}{=} \mathsf{K}_B \mathsf{K}_B^\dagger$ is called the *hat matrix*. It allows us to rewrite the problem as:

$$\min_{\mathsf{S} \in \mathbb{R}^{m \times d}} \|(\mathrm{I} - \hat{\mathsf{K}}_B) \mathsf{K}_S \mathsf{S}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top \mathsf{S} = \mathrm{I}_{(d \times d)} \qquad \text{and} \qquad \mathsf{S}^\top \mathbf{1}_{(m \times 1)} = \mathbf{0}_{(d \times 1)}. \tag{15}$$

As discussed above, the reference-space solution is invariant to similarity transformations. Therefore, the position constraint $\mathsf{S}^\top \mathbf{1} = \mathbf{0}$ can be simply enforced by adding it to the cost, giving the problem:

$$\min_{\mathsf{S}\in\mathbb{R}^{m\times d}} \|\mathsf{LS}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top\mathsf{S} \;=\; \mathrm{I}_{(d\times d)} \qquad \text{with} \qquad \mathsf{L} \;\overset{\text{def}}{=}\; \begin{pmatrix} (\mathrm{I}-\hat{\mathsf{K}}_B)\mathsf{K}_S \\ \mathbf{1}^\top \end{pmatrix} \in \mathbb{R}^{\alpha+1\times m}. \tag{16}$$

A more formal demonstration is given in Appendix A.2. As opposed to the previous problem formulation (12), one could now find the constrained minimizer $\mathsf{S}^\star$ using a simple SVD of matrix L, as proved in appendix A.1. The procedure would be to form L by explicitly using a sparse matrix representation, and compute its d least singular vectors to get $\mathsf{S}^\star$ using some sparse SVD package. Matrix $\hat{\mathsf{K}}_B$ can be formed block-wise since $\mathsf{K}_B = \operatorname{diag}(\mathsf{K}_{B,i})$ implies $\hat{\mathsf{K}}_B = \operatorname{diag}\left(\hat{\mathsf{K}}_{B,i}\right)$. The complexity of computing $\hat{\mathsf{K}}_B$ is thus proportional to the inversion of the symmetric matrices $\mathsf{K}_{B,i}^\top \mathsf{K}_{B,i} \in \mathbb{R}^{d+1\times d+1}$.

The solution we propose only uses regular (full) matrices, and goes deeper into exploiting the problem structure. We first note that the right singular vectors of L and of $\mathsf{W} \overset{\text{def}}{=} \mathsf{L}^\top\mathsf{L}$ are identical. We thus rewrite the problem as:

$$\min_{\mathsf{S}\in\mathbb{R}^{m\times d}} \|\mathsf{WS}\|_{\mathcal{F}}^2 \qquad \text{s.t.} \qquad \mathsf{S}^\top\mathsf{S} \;=\; \mathrm{I}_{(d\times d)} \qquad \text{with} \qquad \mathsf{W} \;\overset{\text{def}}{=}\; \mathsf{L}^\top\mathsf{L} \in \mathbb{R}^{m\times m}. \tag{17}$$

The computational complexity thus depends on the number of model points m but not on the number of shape data n. Matrix $\mathsf{W} \in \mathbb{R}^{m\times m}$ is given by:

$$\mathsf{W} \;=\; \mathbf{1}\mathbf{1}^\top + \sum_{i=1}^{n} \mathsf{K}_{S,i}^\top(\mathrm{I}-\hat{\mathsf{K}}_{B,i})\mathsf{K}_{S,i},$$

since $\left(\hat{\mathsf{K}}_{B,i}-\mathrm{I}\right)^\top\left(\hat{\mathsf{K}}_{B,i}-\mathrm{I}\right) = \mathrm{I}-\hat{\mathsf{K}}_{B,i}$. Matrix $\mathsf{K}_{S,i}$ just inserts rows and columns of zeros. Matrix W can therefore be constructed very efficiently. The method in appendix A.1 is used to find the solution. Our algorithm is given in table 2. Once S is computed from (17), B is simply found from (14) as:

$$\begin{pmatrix} \mathsf{B}_i & \mathbf{b}_i \end{pmatrix} \;=\; -\left(\mathsf{K}_{B,i}^\dagger \mathsf{K}_{S,i}\mathsf{S}\right)^\top.$$

6 Similarity-Euclidean Registration by Upgrading

In this section we deal with the problem of initializing a similarity or a Euclidean registration from an affine registration $A_i = (\mathsf{A}_i; \mathbf{a}_i)$, $i = 1, \ldots, n$. In the Euclidean case, this requires one to find the registration

Objective

Affine registration of multiple matched shapes represented by d-dimensional points, with missing data, using the reference-space model. The result is meant to be used as an initial guess to affine registration with the data-space model of table.

Inputs

The inputs are n matched shapes with at most m points each. Each shape is represented by a matrix D_i, where $i = 1, \ldots, n$ is the shape index. The size of matrix D_i is the number of points observed in shape i times the dimension d (D_i is just like a full shape matrix, for which the rows corresponding to missing points were removed.) Each shape D_i is represented in homogeneous coordinates by the matrix $\bar{\mathsf{D}}_i = (\mathsf{D}_i \;\; \mathbf{1})$. Visible and missing points are indicated by $v_{i,j} \in \{0, 1\}$, where $j = 1, \ldots, m$ is the point index.

Outputs

The outputs are the n affine transformations $T_{A,i} = (\mathsf{A}_i; \mathbf{a}_i) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times 1}$ and the reference shape in matrix $\mathsf{S} \in \mathbb{R}^{m\times d}$ with $\mathsf{S}^\top = (\mathbf{S}_1 \;\; \cdots \;\; \mathbf{S}_m)$.

Algorithm

```
• For $i = 1, \ldots, n$, Set $\mathsf{K}^\dagger_{B,i} \leftarrow \left(\bar{\mathsf{D}}_i^\top \bar{\mathsf{D}}_i\right)^{-1} \bar{\mathsf{D}}_i^\top$, Set $\hat{\mathsf{K}}_{B,i} \leftarrow \bar{\mathsf{D}}_i \mathsf{K}^\dagger_{B,i}$, Endfor
• Set $\mathsf{W} \leftarrow \mathsf{1}_{(m\times m)}$
• For $i = 1, \ldots, n$                    $\{ \mathsf{W} {+}{=} \sum_{i=1}^n \mathsf{K}_{S,i}^\top (\mathrm{I} - \hat{\mathsf{K}}_{B,i}) \mathsf{K}_{S,i} \}$
   Set $k_1 \leftarrow 1$
   For $j_1 = 1, \ldots, m$
     If $v(i, j_1)$
       Set $k_2 \leftarrow 1$
       For $j_2 = 1, \ldots, m$
         If $v(i, j_2)$
           If $j_1 = j_2$, $\mathsf{W}(j_1, j_2){+}{=}1$, Endif
           Set $\mathsf{W}(j_1, j_2){-}{=}\hat{\mathsf{K}}_{B_i}(k_1, k_2)$
           Set $k_2 \leftarrow k_2 + 1$
         Endif
       Endfor
       Set $k_1 \leftarrow k_1 + 1$
     Endif
   Endfor
  Endfor
• Compute an SVD $\mathsf{W} \overset{\mathsf{SVD}}{\longrightarrow} \mathsf{U}\boldsymbol{\Sigma}\mathsf{U}^\top$
• Set $\mathsf{S}^\star$ to the last $d$ columns of $\mathsf{U}$
• For $i = 1, \ldots, n$
   Set $\mathsf{B}_i \leftarrow \mathsf{0}_{(d\times d+1)}$
   Set $k \leftarrow 1$
   For $j = 1, \ldots, m$                    $\{ \mathsf{B}_i \leftarrow -(\mathsf{K}^\dagger_{B,i} \mathsf{K}_{S,i} \mathsf{S})^\top \}$
     If $v(i, j)$
       For $d_1 = 1, \ldots, d+1$, For $d_2 = 1, \ldots, d$, Set $\mathsf{B}_i(d_2, d_1){-}{=}\mathsf{K}^\dagger_{B,i}(k, d_1)\mathsf{S}(j, d_2)$, Endfor, Endfor
       Set $k \leftarrow k + 1$
     Endif
   Endfor
  Endfor
• For $i = 1, \ldots, n$                    { Extracting the forward transformations }
   Set $\mathsf{A}_i \leftarrow \mathsf{B}_i^{-1}$
   Set $\mathbf{a}_i \leftarrow -\mathsf{A}_i \mathbf{b}_i$
  Endfor
```

Table 2: **Affine registration with the reference-space model.** Implementation of our initial closed-form affine registration algorithm in the reference-space model. The mathematical derivation is in §5.

$\mathcal{E}_i = (\mathsf{E}_i; \mathbf{e}_i)$ such that $\mathsf{E}_i \in SO(d)$, $i = 1, \ldots, n$. A trivial solution that comes to mind is to find each E_i by projecting A_i to the closest orthonormal matrix. This however cannot be done as directly. Indeed, one has to take into account the nature of the gauge. In the data-space model framework, we have seen that the gauge is modeled by a transformation of the same nature as the registration that is to be computed. In other words, the affine registration we compute is up to an unknown global affine transformation $G = (\mathsf{G}; \mathbf{g})$, that we have to resolve (at least partly) so as to upgrade the registration to similarity-Euclidean. The transformation upgrading equation is thus $A \rightarrow A \circ G^{-1}$, and G is dubbed the *upgrade transformation.* We start by expanding it to:

$$(\mathsf{A}_i; \mathbf{a}_i) \;\rightarrow\; (\mathsf{A}_i\mathsf{G}^{-1}; -\mathsf{A}_i\mathsf{G}^{-1}\mathbf{g} + \mathbf{a}_i) \;\approx\; (\mathsf{E}_i; \mathbf{e}_i) \qquad \text{with} \qquad \mathsf{E}_i \in SO(d),\;\; i = 1, \ldots, n.$$

Only the rotational part provides constraints on the upgrade transformation. The translational part is chosen as $\mathbf{g} = \mathbf{0}$, and we thus get $\mathbf{e}_i = \mathbf{a}_i$, $i = 1, \ldots, n$. The Euclidean upgrading problem is thus formulated as:

$$\operatorname*{find}_{\mathsf{G},\mathsf{E}_1,\ldots,\mathsf{E}_n} \qquad \text{s.t.} \qquad \mathsf{A}_i\mathsf{G}^{-1} \;\approx\; \mathsf{E}_i \qquad \text{with} \qquad \mathsf{E}_i \in SO(d),\;\; i = 1, \ldots, n.$$

Using the decomposition $\mathsf{G} = \mathsf{QZ}$ of equation (8), where $\mathsf{Q} \in SO(d)$ and Z is an upper triangular matrix, we can eliminate the rotational part Q of the upgrade transformation to get:

$$\operatorname*{find}_{\mathsf{Z},\mathsf{E}_1,\ldots,\mathsf{E}_n} \qquad \text{s.t.} \qquad \mathsf{A}_i\mathsf{Z}^{-1} \;\approx\; \mathsf{E}_i \qquad \text{with} \qquad \mathsf{E}_i \in SO(d),\;\; i = 1, \ldots, n.$$

This problem does not have an exact solution. We solve it in two steps. First, we compute the upgrade transformation Z. Second, we find the sought transformations E_i, $i = 1, \ldots, n$. We note that this formulation also holds but up to scale in the similarity case:

$$\operatorname*{find}_{\mathsf{Z},\mathsf{E}_1,\ldots,\mathsf{E}_n,\zeta_1,\ldots,\zeta_n} \qquad \text{s.t.} \qquad \mathsf{A}_i\mathsf{Z}^{-1} \;\approx\; \zeta_i\mathsf{E}_i \qquad \text{with} \qquad \mathsf{E}_i \in SO(d),\;\; \zeta_i > 0,\;\; i = 1, \ldots, n.$$

Note that in this case one can also impose a constraint on the scale of the upgrading – this will be done during the first step of the algorithm. Our algorithm is summarized in table 3.

6.1 Computing the Upgrading Transformation

This problem looks similar to the one of self-calibration in Structure-from-Motion for the affine camera (Quan, 1996). We follow the classical approach of discarding the shape and using the affine registration parameters only to find the upgrading transformation. However, this leads to a much simpler and more stable algorithm than in the Structure-from-Motion case. We will also sketch an alternative solution using the shape to find the upgrading transformation, and will show that the two solutions are equivalent. We tackle the Euclidean case first and then show how our solution can be extended to the similarity case.

Euclidean case. The rotational part must be special orthonormal: $\mathsf{A}_i\mathsf{Z}^{-1} \in SO(d)$, giving the following orthonormality constraints:

$$\mathsf{A}_i\mathsf{Z}^{-1}\mathsf{Z}^{-\top}\mathsf{A}_i^\top \;\approx\; \mathrm{I} \qquad \text{and} \qquad \det(\mathsf{A}_i\mathsf{Z}^{-1}) \;>\; 0,\;\; i = 1,\ldots,n.$$

The affine registration does not make reflections between the inter-shape transformations; in other words, it is consistently oriented. It means that $\mathrm{sign}(\det(\mathsf{A}_1)) = \cdots = \mathrm{sign}(\det(\mathsf{A}_n)) = \omega \in \{-1;1\}$. Since $\det(\mathsf{A}_i\mathsf{Z}^{-1}) = \det(\mathsf{A}_i)\det(\mathsf{Z}^{-1})$, the constraints simply become $\omega\det(\mathsf{Z}^{-1}) > 0$. This single constraint can further be rewritten as $\mathrm{sign}(\det(\mathsf{Z})) = \omega$ leading to the following set of constraints:

$$\begin{aligned} \mathsf{A}_i\mathsf{Y}\mathsf{A}_i^\top &\approx \mathrm{I},\;\; i = 1,\ldots,n && (18)\\ \mathsf{Y} &= \mathsf{Z}^{-1}\mathsf{Z}^{-\top} && (19)\\ \mathrm{sign}(\det(\mathsf{Z})) &= \omega, && (20)\end{aligned}$$

In stratified self-calibration Y represents the Dual Absolute Conic, that contains the metric structure of the 3D model (Hartley and Zisserman, 2003). In stratified generalized procrustes analysis Y also represents a second-order surface in the reference-space and contains the Euclidean structure of the reference shape (or the metric structure in the similarity case.) The problem directly above can be easily cast as a linear least squares problem with a cost given by equation (18) under the bilinear equality constraints of equation (19) and the degree-d inequality constraint of equation (20). Because they use projection matrices, linear self-calibration algorithms use equation (18) only to solve for Y from the A_i, $i = 1,\ldots,n$ (Pollefeys et al., 2004). Constraint (19) can be enforced *a posteriori* (Gurdjos et al., 2009). This is known to be a difficult and sometimes unstable step.

We now show how to further rearrange the system of constraints (18), (19) and (20) so that all of them can be satisfied by a simple linear least squares estimator. The main idea is, as in linear self-calibration, to first compute Y and then extract Z based on equation (19) (using for instance the inverse of a Cholesky factor.) We first show that constraint (20) can be ignored when solving for Y. If the solution $\mathsf{Z}^\star$ that is extracted from Y does not satisfy (20) we can simply flip the sign of its last entry: $Z^\star_{d,d} \leftarrow -Z^\star_{d,d}$. This way both (19) and (20) will be satisfied. We now show that, rearranging (18) using the fact that the A_i,

$i = 1, \ldots, n$ are square invertible matrices, (19) (which merely is $\mathsf{Y} \in \mathbb{S}(d)$ *i.e.*, that Y must be symmetric positive definite) can be ignored as well. Rewriting (18) as:

$$\mathsf{Y} \;\approx\; \mathsf{A}_i^{-1}\mathsf{A}_i^{-\top}, \;\; i = 1, \ldots, n,$$

we get the following constrained linear least squares problem:

$$\min_{\mathsf{Y} \in \mathbb{S}(d)} \sum_{i=1}^{n} \|\mathsf{Y} - \mathsf{A}_i^{-1}\mathsf{A}_i^{-\top}\|_{\mathcal{F}}^2,$$

whose solution is simply:

$$\mathsf{Y}^\star \;\leftarrow\; \frac{1}{n}\sum_{i=1}^{n} \mathsf{A}_i^{-1}\mathsf{A}_i^{-\top}.$$

It is easily shown that $\mathsf{Y}^\star$ satisfies all the necessary constraints. In practice, noting that the Cholesky factor of $\mathsf{Y}^\star$ have to be inverted to find $\mathsf{Z}^\star$, we directly compute $\mathsf{P} \stackrel{\text{def}}{=} \mathsf{Y}^{-1}$ and obtain $\mathsf{Z}^\star$ as the Cholesky factor of $\mathsf{P}^\star$ to get the closed-form solution:

$$\mathsf{Z}^\star \;\leftarrow\; \mathrm{chol}\left(\frac{1}{n}\sum_{i=1}^{n} \mathsf{A}_i^{\top}\mathsf{A}_i\right).$$

Similarity case. In the similarity case the constraint (18) becomes $\mathsf{A}_i\mathsf{Y}\mathsf{A}_i^\top \approx \xi_i \mathrm{I}$, $i = 1, \ldots, n$, for some unknown $\xi_i > 0$ defined as the square of the transformation scale: $\xi_i \stackrel{\text{def}}{=} \zeta_i^2$. We fix the determinant of Y to be some positive constant c to set the otherwise free global scale of the gauge. Defining $\boldsymbol{\xi}^\top \stackrel{\text{def}}{=} (\xi_1 \;\; \cdots \;\; \xi_n)$, we get the constraints:

$$\begin{aligned}
\mathsf{A}_i\mathsf{Y}\mathsf{A}_i^\top &\approx \xi_i \mathrm{I}, \;\; i = 1, \ldots, n && (21)\\
\xi_i &> 0, \;\; i = 1, \ldots, n && (22)\\
\det(\mathsf{Y}) &= c \; > \; 0 && (23)\\
\mathsf{Y} &= \mathsf{Z}^{-1}\mathsf{Z}^{-\top} && (24)\\
\mathrm{sign}(\det(\mathsf{Z})) &= \omega. && (25)
\end{aligned}$$

We now propose a solution that will exactly satisfy the constraints directly above using a simple linear least squares estimator, as in the simplest Euclidean case. We first rewrite the set of constraints using $\mathsf{P} = \mathsf{Y}^{-1}$

Objective

Upgrading an affine generalized procrustes analysis to similarity or Euclidean generalized procrustes analysis. The result is meant to be used as an initial guess for nonlinear similarity-Euclidean registration.

Inputs

The inputs are n affine transformations $(\mathsf{A}_i; \mathbf{a}_i)$, $i = 1, \dots, n$ in dimension d. These transformations must have a consistent orientation: they must satisfy $\omega = \mathrm{sign}(\det(\mathsf{A}_1)) = \cdots = \mathrm{sign}(\det(\mathsf{A}_n))$.

Outputs

In the Euclidean-upgrading case, the outputs are n Euclidean transformations $(\mathsf{E}_i, \mathbf{e}_i)$ with $\mathsf{E}_i \in SO(d)$, $i = 1, \dots, n$. In the similarity-upgrading case, the algorithm also outputs the transformation scales $\zeta_i > 0$, $i = 1, \dots, n$. If the reference shape S is known, it is only upgraded.

Algorithm

```
• Computing the upgrade transformation
  Euclidean case:
      $$\text{Set } \mathsf{Z} \;\leftarrow\; \mathrm{chol}\left(\frac{1}{n}\sum_{i=1}^{n} \mathsf{A}_i^\top \mathsf{A}_i\right)$$
  Similarity case:
      $$\text{Set } \mathsf{Z} \;\leftarrow\; \mathrm{chol}\left(\sum_{i=1}^{n} \sqrt[d]{\frac{1}{\det(\mathsf{A}_i^\top \mathsf{A}_i)}}\mathsf{A}_i^\top \mathsf{A}_i\right)$$
• $Z_{d,d} \leftarrow \omega\, \mathrm{sign}(\det(\mathsf{Z}))$
• Applying the upgrade transformation
   For $i = 1, \dots, n$
     Set $\mathbf{e}_i \leftarrow \mathbf{a}_i$
     Set $\mathsf{A}_i \mathsf{Z}^{-1} \overset{\mathsf{SVD}}{\rightarrow} \mathsf{U}\Sigma\mathsf{V}^\top$
     Set $\mathsf{E}_i \leftarrow \mathsf{U}\mathsf{V}^\top$
     Set $\zeta_i \leftarrow \frac{1}{d}\sum_{k=1}^{d} \sigma_k$ (similarity case)
   Endfor
   Set $\mathsf{S} \leftarrow \mathsf{S}\mathsf{Z}^\top$
```

Table 3: **Similarity-Euclidean registration by upgrading an affine registration**. Implementation of our closed-form upgrading algorithm. The mathematical derivation is in §6.

and $\phi_i \overset{\text{def}}{=} \xi_i^{-1}$ as follows:

$$\begin{aligned}
\mathsf{A}_i \mathsf{P}^{-1} \mathsf{A}_i^\top &\approx \phi_i^{-1}\mathrm{I}, \;\; i = 1, \ldots, n && (26)\\
\phi_i &> 0, \;\; i = 1, \ldots, n && (27)\\
\det(\mathsf{P}) &= b \;>\; 0 && (28)\\
\mathsf{P} &= \mathsf{Z}^\top \mathsf{Z} && (29)\\
\mathrm{sign}(\det(\mathsf{Z})) &= \omega, && (30)
\end{aligned}$$

with $b \overset{\text{def}}{=} \frac{1}{c}$. We start using the same arguments as in the Euclidean case and drop constraint (30) that will be enforced *a posteriori* when Z will be extracted from the computed P as a Cholesky factor. Defining $\mathsf{F}_i = \mathsf{A}_i^\top \mathsf{A}_i$, $i = 1, \ldots, n$ we then proceed to rewrite the constraint (26) as:

$$\mathsf{P} \;\approx\; \phi_i \mathsf{F}_i, \;\; i = 1, \ldots, n.$$

Taking the determinant of both sides of the equation gives:

$$\det(\mathsf{P}) \;\approx\; \phi_i^d \det(\mathsf{F}_i),$$

and the solution:

$$\phi_i^\star \;=\; \sqrt[d]{\frac{b}{\det(\mathsf{F}_i)}}.$$

Since $\det(\mathsf{F}_i) > 0$ we get that $\phi_i^\star > 0$, $i = 1, \ldots, n$ as required. Finally, the least squares solution for P is obtained as:

$$\mathsf{P}^\star \;\leftarrow\; \sum_{i=1}^{n} \phi_i^\star \mathsf{F}_i.$$

It is to be observed that with this expression, ϕ_i, $i = 1, \ldots, n$ implies that $\mathsf{P}^\star \in \mathbb{S}(d)$. In other words, enforcing the constraint (27) will as a consequence enforce the constraint (29). The value of b can be chosen arbitrarily (in practice we choose $b = 1$.) The constraint $\det(\mathsf{P}^\star) = b$ can be enforced by simply rescaling $\mathsf{P}^\star$, but it is not necessary since b represented the arbitrary global scale. The whole set of constraints (26)-(30) can thus be enforced by a simple linear least squares estimate.

A derivation based on the reference shape. Instead of using the affine transformations as in the previous self-calibration-like derivation, we propose to keep the affine shape S. What we know is that there exists an upgrade transformation Z and orthonormal transformations E_i such that:

$$\mathbf{D}_{i,j} \approx \mathsf{E}_i \mathsf{Z} \mathbf{S}_j + \mathbf{a}_i.$$

Let $\mathcal{X} \in \mathbb{R}^{nd \times m}$ be the centred measurement matrix,[6] with entries $\mathbf{D}_{i,j} - \mathbf{a}_i$. We may rewrite the previous equation in matrix form as:

$$\mathcal{X} \approx \mathcal{R}\mathsf{Z}\mathsf{S}^\top \qquad \text{with} \qquad \mathcal{X} \overset{\text{def}}{=} \begin{pmatrix} \mathbf{D}_{1,1} - \mathbf{a}_1 & \cdots & \mathbf{D}_{1,m} - \mathbf{a}_1 \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{n,1} - \mathbf{a}_n & \cdots & \mathbf{D}_{n,m} - \mathbf{a}_n \end{pmatrix} \qquad \text{and} \qquad \mathcal{R} \overset{\text{def}}{=} \begin{pmatrix} \mathsf{E}_1 \\ \vdots \\ \mathsf{E}_n \end{pmatrix}.$$

Given that S is column-orthonormal, $\mathsf{S}^\top\mathsf{S} = \mathrm{I}$. We multiply each side of the equation to the right by S and get:

$$\mathcal{X}\mathsf{S} \approx \mathcal{R}\mathsf{Z}.$$

Since $\mathcal{R}$ is made of orthonormal matrices, we have that $\mathcal{R}^\top\mathcal{R} = \mathrm{I}$. Multiplying each side of the equation to the left by its transpose gives:

$$\mathsf{S}^\top\mathcal{X}^\top\mathcal{X}\mathsf{S} \approx \mathsf{Z}^\top\mathsf{Z} = \mathsf{P}. \tag{31}$$

As in the previous solution we can compute P and then Z as a Cholesky factor of P. This solution is simple and uses the data points instead of an algebraic criterion. However, one may wonder how different are the two solutions. To answer this question, we consider the maximum likelihood transformation A_i given by:

$$\mathsf{A}_i = \min_{\mathsf{A}_i \in \mathbb{R}^{d\times d}} \left\| \mathsf{D}_i^\top - \mathsf{A}_i\mathsf{S}^\top - \mathbf{a}_i\mathbf{1}^\top \right\|_{\mathcal{F}}^2 = \left(\mathsf{D}_i^\top - \mathbf{a}_i\mathbf{1}^\top\right)\mathsf{S}.$$

The solution (31) can thus be rewritten as:

$$\mathsf{S}^\top\mathcal{X}^\top\mathcal{X}\mathsf{S} = \sum_{i=1}^{n} \mathsf{S}^\top\left(\mathsf{D}_i - \mathbf{1}\mathbf{a}_i^\top\right)\left(\mathsf{D}_i^\top - \mathbf{a}_i\mathbf{1}^\top\right)\mathsf{S} = \sum_{i=1}^{n} \mathsf{A}_i^\top\mathsf{A}_i,$$

which shows that the two solutions are the same provided that the affine registration is maximum likelihood in the data-space model. Note that this also holds for the similarity case.

[6] So as to make the approach usable even if data points are missing, one uses a *partly predicted* centred measurement matrix built by using for the missing data points their prediction from the reference shape and affine registration: $\mathbf{D}_{i,j} \approx \mathsf{A}_i\mathbf{S}_j + \mathbf{a}_i$.

6.2 Applying the Upgrading Transformation

Once the upgrading transformation Z is computed, one can then find the similarity-Euclidean transformations. This implies that one solves the so-called 'orthonormal correction' step, that is similar to classical procrustes analysis of a pair of shape data. The solution is thus well-known.

We sought an orthonormal transformation $\mathsf{E}_i \in SO(d)$ 'close to' $\mathsf{A}_i\mathsf{Z}^{-1}$. A classical solution to this problem is to find E_i such that:

$$\mathsf{E}_i \; = \; \arg \min_{\mathsf{E}_i \in SO(d)} \|\mathsf{E}_i - \mathsf{A}_i\mathsf{Z}^{-1}\|^2_{\mathcal{F}}.$$

In the similarity case, one also has to find a scale factor $\zeta_i > 0$ leading to:

$$(\mathsf{E}_i; \zeta_i) \; = \; \arg \min_{\substack{\mathsf{E}_i \in SO(d) \\ \zeta_i > 0}} \|\zeta_i\mathsf{E}_i - \mathsf{A}_i\mathsf{Z}^{-1}\|^2_{\mathcal{F}}.$$

It is known that the solution to this problem is $\mathsf{E}_i \leftarrow \mathsf{U}\mathsf{V}^\top$ where $\mathsf{A}_i\mathsf{Z}^{-1} \overset{\mathsf{SVD}}{\to} \mathsf{U}\Sigma\mathsf{V}^\top$ is an SVD and ξ_i is the average of the singular values: $\xi_i \leftarrow \frac{1}{d}\left(\sum_{k=1}^{d} \sigma_k\right)$ where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_d)$. A demonstration is given in Appendix A.3.

7 Experimental Results

We present experimental results comparing the proposed methods to existing ones using simulated and real data and for various dimensions.

7.1 Compared Methods and Error Criterion

Each method being compared has a shortname in two parts. It first indicates the group of transformation being computed (Aff for affine, Sim for similarity and Euc for Euclidean.) The second part of a method's shortname indicates the nature of the method itself (the algorithm being used.) The compared methods are:

- Aff-Fct – affine factorization (for complete datasets only)
- Aff-Ref – the proposed affine closed-form reference-space solution described in table 2
- Aff-All – affine refinement in data-space
- Sim-Alt/Euc-Alt – the classical alternation method described in §3
- Sim-Upg/Euc-Upg – the proposed upgrading to similarity/Euclidean described in table 3

- Sim-All/Euc-All – similarity/Euclidean refinement in data-space

The error criterion we measure is the data-space cost (4). We also measure the computation time.

7.2 Simulated Data

We first describe our simulation setup and then report our experimental results. The results are obtained as RMS (Root Mean of Squares) over 100 random trials with the same simulation parameters.

7.2.1 Simulation Setup

We simulate shape data using a random reference shape and transformations. We here recall that our input data are a set of shape data. Each shape datum is a set of points called shape points. Our simulation setup has several parameters that we will vary in turn to assess the algorithms' behaviour in various conditions. Our description below includes the default value and range of variation of the simulation parameters. For instance, $m(50; [10, 50])$ means that the number m of simulated shape points is 50 by default and will vary from 10 to 50 in our experiments. Those m reference shape points are randomly drawn in an origin-centred hyper-sphere of unit radius in dimension $d(3; [1, 10])$. We then generate $n(5; [2, 50])$ affine transformations from $d+1$ control points each. We proceed by setting the control points in the reference coordinate frame as the d canonical basis vectors and the origin. We then draw the n sets of control points at random in the unit hyper-sphere and form the n sought affine transformations. So as to generate Euclidean transformations we perform a QR decomposition as in equation (8) on the rotational part of each affine transformation and keep the orthonormal part and the scale of the purely affine part only. Finally, we apply the n transformations to the m reference shape points. We simulate noise and nonrigidity by adding normally distributed random values with standard deviation σ. For rigid simulation we use lower $\sigma^2(.01; [0, .1])$ values and interpret it as noise. For nonrigid simulation we use higher values $\sigma^2(.1; [0, 1])$ which can be thought of as deviation from the estimated model, equivalent to nonrigid transformations applied to the shape points. We finally simulate missing data by erasing some of the generated shape points with rate $\tau(.5; [0, .7])$ (.5 means 50% missing data.)

7.2.2 Affine Generalized Procrustes Analysis

The experiments we report in this section are meant to assess the quality of the affine part in our stratified approach and thus will compare Aff-Fct, Aff-Ref and Aff-All.

The first set of experiments is intented to check if Aff-Ref followed by Aff-All manages to find the optimal solution. We use Aff-Fct, which always finds the global minimum, as a reference, for complete datasets ($\tau = 0$.) The results are shown in the four top rows of figure 4. We will simulate missing data in a second step; however Aff-All means reference-space initialization followed by iterative minimization of the data-space error. We simulated three kinds of data: Euclidean, affine and non-rigid.

In the Euclidean data case shown in the left column of figure 4 we observe that all three methods gently degrade with increasing dimension d and noise σ while they are steady with respect to the number of points m and number of shapes n. The error decreases with respect to the amount of missing data τ which is logical since less data means less constraints to be satisfied to fit the data. What is very important in this first batch of experiments is that, as will be confirmed by the other experiments, Aff-All always find the optimal solution like Aff-Fct and that, more importantly, Aff-Ref gives a very close result in all cases. This is very important since Aff-Ref is based on a closed-form solution. This can easily be explained: Aff-Ref minimizes a reference-space error while the other two methods minimize a data-space error; with Euclidean data both errors are very similar, and so the global solution to one closely matches the global solution to the other.

In the affine data case shown in the middle column of figure 4 we first observe that Aff-All and Aff-Fct always reach the same error. This means that Aff-All always finds the global minimum. The second observation is about Aff-Ref. The results that this method obtains can be very similar or different to the results obtained by the two other methods. Remember that Aff-Ref solves a convex approximation of the original problem and is thus not supposed to reach the exact solution to the original problem. The solution it finds is however not too far from the optimal one in general. It gently degrades with the dimension d and the number of shape data n increasing but is about steady regarding the number of points m. It however quickly degrades beyond a noise standard deviation σ of about 0.2 (20% of the shapes' size) which is way beyond typical real noise levels.

In the non-rigid data case shown in the right of figure 4 we can make similar observations as in the rigid case regarding methods Aff-All and Aff-Fct: they both always converge to the global minimum. The result of Aff-Ref however quickly degrades with respect to all parameters and could probably not be used 'as is'; it is sufficiently reliable however to allow Aff-All to converge to the global minimum.

In figure 4, we also present results obtained when varying the amount of missing data in the rigid and nonrigid cases. The results are shown in the two right-most graphs of figure 4. In this setup we cannot run Aff-Fct since this factorization based method does not cope with missing data; only Aff-All and Aff-Ref are thus compared. We observe that Aff-Ref resists to missing data quite well.

As a conclusion, we can say that Aff-Ref provides an initialization that allows Aff-All to reach the global minimum of the cost function in all cases. This solution is extremely accurate for Euclidean data,

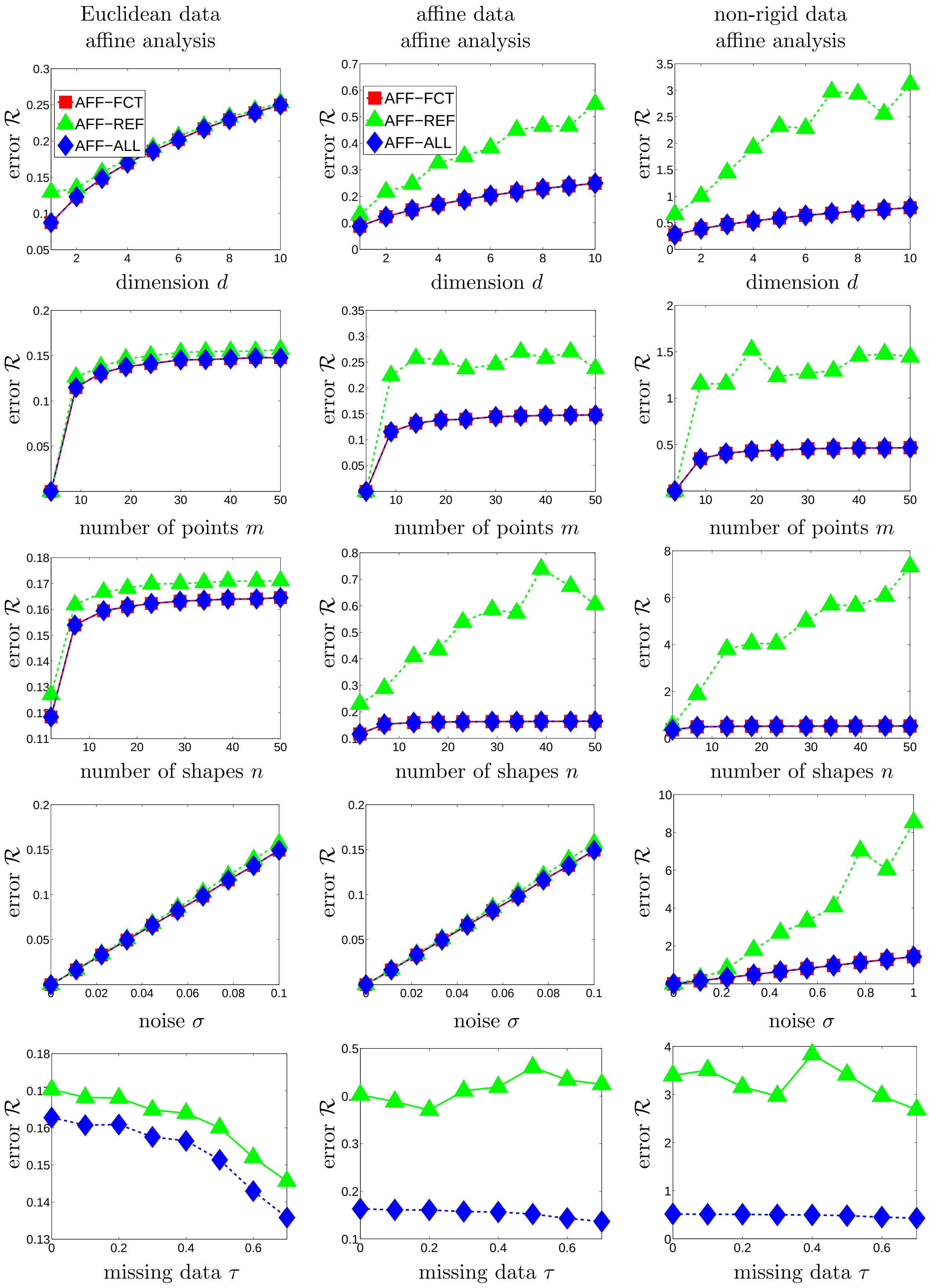


Figure 4: **Affine generalized procrustes analysis with simulated data.** The graphs show the error as a function of 5 different parameters. Each column is for a different simulation setup; from left to right: Euclidean, affine and non-rigid transformations, as indicated at the top.

which is a very important case in practice in 2D and 3D shape analysis.

7.2.3 Similarity and Euclidean Generalized Procrustes Analysis

Our experiments in this section are meant to assess the quality of the similarity and Euclidean part in our stratified approach and thus will compare Sim-Upg, Sim-All and Sim-Alt, and the Euclidean equivalents. Our goal is to see how our algorithms behave with respect to various factors, and how they compare to classical alternation. Note that only dimensions $d \in \{2, 3\}$ are tested. The experiments intend to check if our algorithm Sim-All improves on the classical alternation Sim-Alt and the capability of Sim-Upg to provide a reliable initialization.

For similarity data we show the result of similarity analysis in the left column of figure 5; very similar results were obtained for Euclidean analysis and we thus do not include the corresponding graphs. We observe that the results of Sim-All and Sim-Alt are close, but that Sim-All always gives a more accurate result. The upgrading method Sim-Upg degrades slightly from dimension 2 to 3 and gently increases its error against the number of shapes and noise, while it gets more accurate as the number of points increases. The middle column of figure 5 shows the results obtained for similarity analysis of non-rigid data. We can draw similar observations as in the rigid case. We also notice that Sim-Upg is very sensitive to the amount of non-rigidity in the shape data. The results for non-rigid data and Euclidean analysis are shown in the right column of figure 5. We make the same observations as for similarity analysis except that the error stabilizes with increasing number of shapes instead of growing.

As a conclusion, we can say that Sim-Upg provides a good initialization to Sim-All. Classical alternation implemented by Sim-Alt performs nearly as well as Sim-All but is always less accurate. The same conclusion holds for Euclidean generalized procrustes analysis.

7.2.4 Timing and Complexity

We monitored the average run-time for each method over the 100 random trials of each experiment. Note that all iterative methods (alternation methods and nonlinear refinement with Orthogonal Distance Regression) use the same termination criterion, based on thresholding the norm of the difference between two consecutive estimates of the reference shape matrix. More specifically, the algorithms stop if this norm is lower than 10^{-6}. We show the timings for the minimum and maximum value of each parameter in figure 6. In the affine analysis case, Aff-All is generally the slowest method while Aff-Fct is always the fastest. Aff-Ref lies inbetween. For similarity and Euclidean analysis and similarity data Sim-Upg and Sim-All (respectively Euc-Upg and Euc-All) are the fastest two methods while the alternation method Sim-Alt (respectively

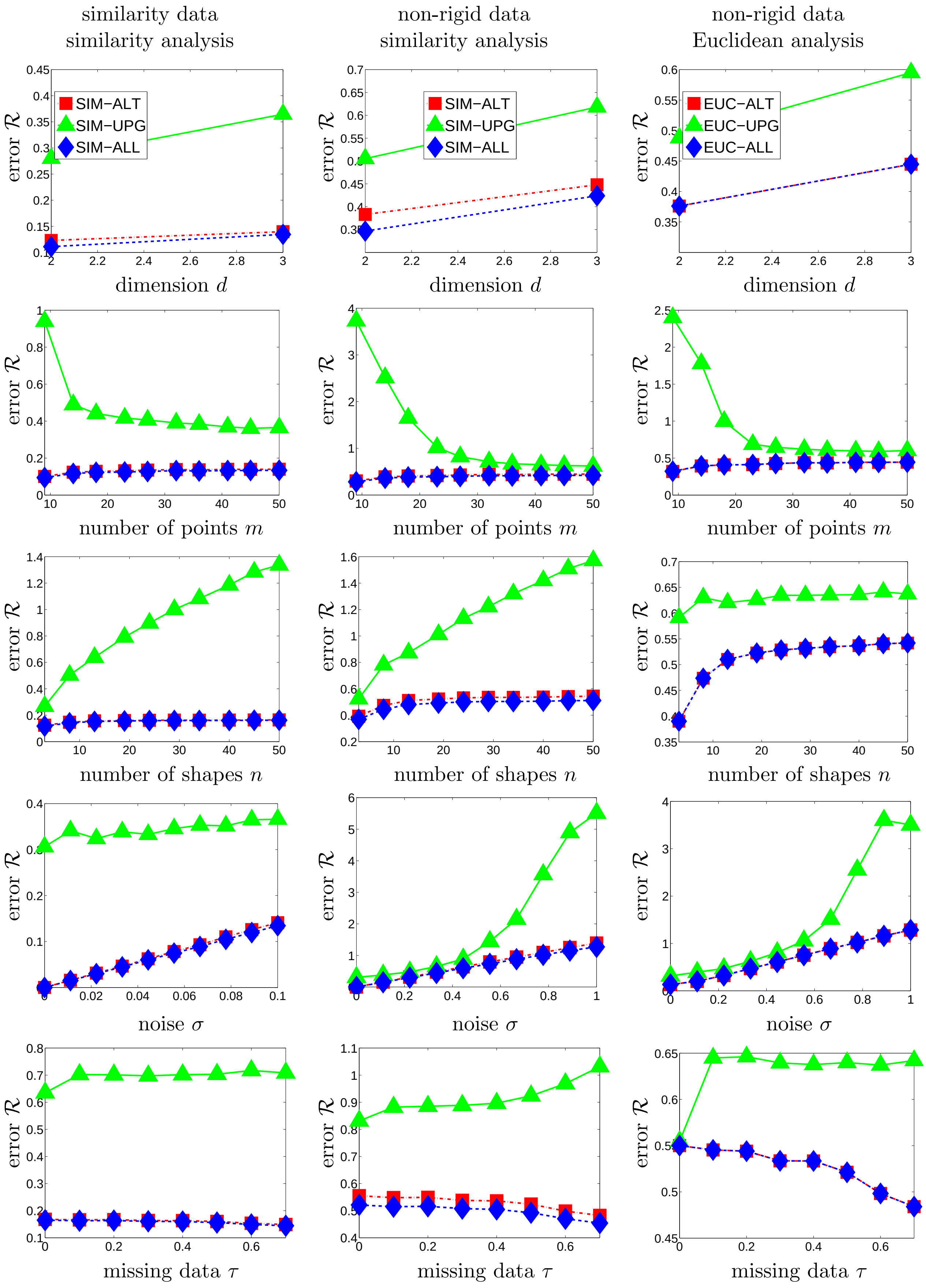


Figure 5: **Similarity and Euclidean generalized procrustes analysis with simulated data.** The graphs show the error as a function of 5 different parameters. Each column is for a different simulation setup and type of transformation, as indicated on the top. For similarity data, similarity and Euclidean generalized procrustes analysis give similar results so we only show the results for similarity generalized

Euc-Alt) is generally the slowest. For non-rigid data however Sim-All and Euc-All are generally the slowest methods; Euc-Alt is generally the fastest and Euc-Upg lies inbetween while Sim-Alt and Sim-Upg are nearly equivalent, Sim-Upg being slighly faster. These timings were monitored with our Matlab implementation of the algorithms; they could of course be substantially different for another implementation, especially in a programming language such as C or C++. They nonetheless give an idea of the required computation time for various cases. As a conclusion, we can say that the nonlinear refinement methods Aff-All, Sim-All and Euc-All are generally the slowest in our implementation. The affine factorization method Aff-Fct is very fast while the timings needed by the alternation methods Sim-Alt and Euc-Alt depend on the type of data.

We now give an analysis of the theoretical complexity of the algorithms. Let κ be the number of iterations performed by an iterative algorithm. Alternation methods have a complexity in $\mathcal{O}(\kappa n d^3)$, assuming that solving for a single transformation has complexity $\mathcal{O}(d^3)$. Aff-Ref has a complexity in $\mathcal{O}(m^3)$, since the SVD has the highest complexity. Upgrading methods have a complexity in $\mathcal{O}(d^3)$, since the Cholesky decomposition has the highest complexity. Finally, Orthogonal Distance Regression methods have a complexity in $\mathcal{O}(\kappa m(nd)^3)$, assuming that sparsity is exploited over the reference shape. It is thus clear that in our proposed stratified framework, the complexity is dominated by Orthogonal Distance Regression. Therefore, the theoretical complexity of our stratified framework is $\mathcal{O}(\kappa m(nd)^3)$, higher than the complexity of alternation methods which is $\mathcal{O}(\kappa n d^3)$. This confirms the conclusions we drew from the timing results.

7.2.5 Influence of the Initialization

We assessed to which extent Aff-Ref is beneficial to Aff-All as an initialization procedure for the nonlinear refinement. For this reason, we compared for all our experiments the nonlinear refinement initialized by Aff-Ref, initialized by a random solution and by a simple combination of the shape data (see below.) Those three methods are respectively denoted Aff-All, Aff-All-Rnd and Aff-All-Smp. We monitored two quantities: the error $\mathcal{R}$ at convergence and the number of iterations required to reach convergence. The initialization in Aff-All-Smp proceeds in two steps. Visible points are used to establish pairwise affine transformations between the shape data. After transfer to some reference coordinate frames, all visible points are then averaged to get the reference shape. Regarding the number of iterations, we obtained the following results:

- Aff-All (refinement started from the solution of Aff-Ref): 7.46 iterations on average, standard deviation 15.43

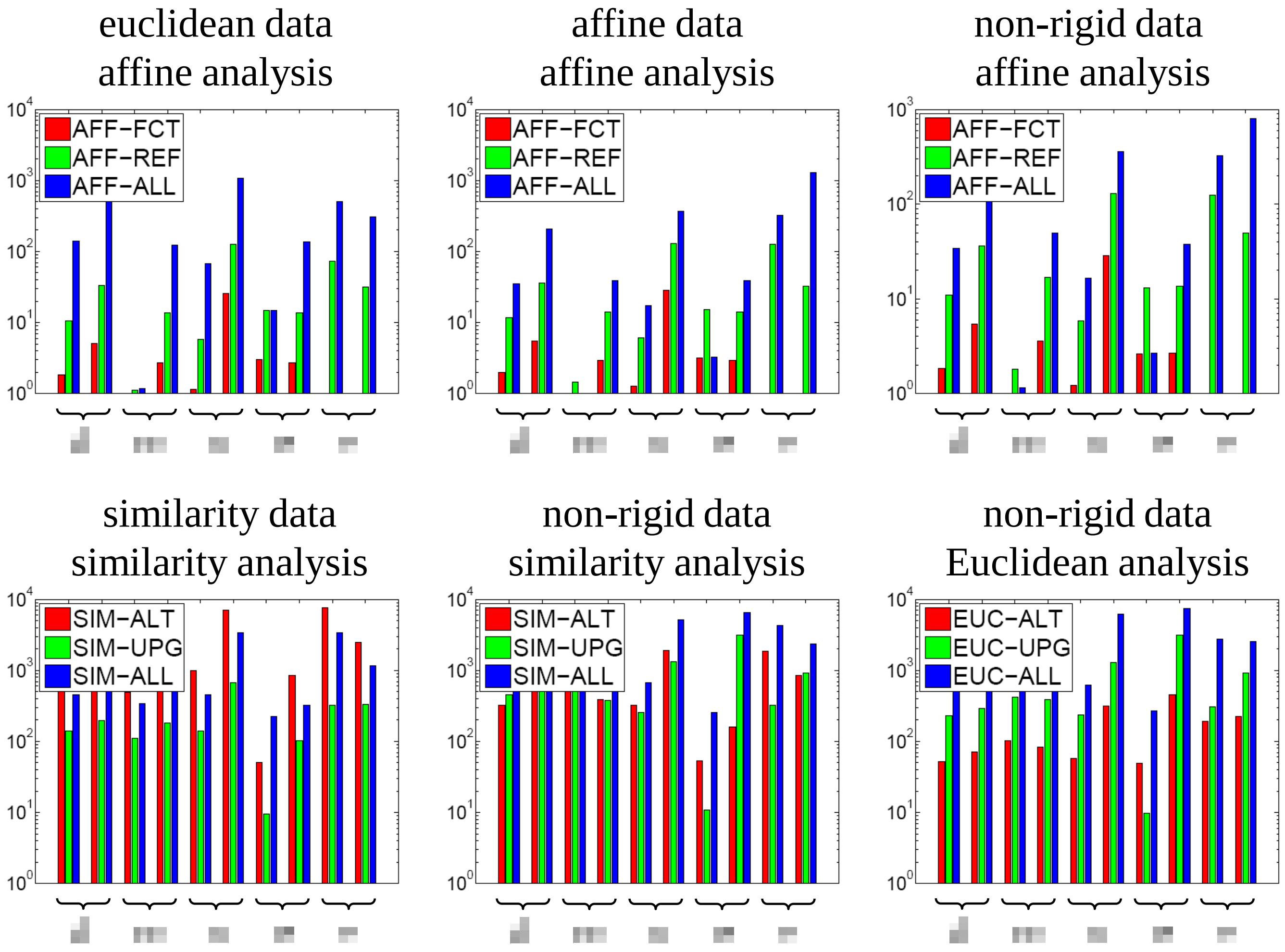


Figure 6: **Timing results for all experiments on simulated data.** For each simulation and analysis scenario shown in one of figures 4 and 5 the timing (in ms) is shown for each of the 5 varied parameters and each tested method for the two extreme parameter values: the timing for the minimum and maximum value of the parameters are shown on the left and right batch of bars respectively.

- Aff-All-Rnd (refinement started from a random solution): 19.15 iterations on average, standard deviation 38.88
- Aff-All-Smp (refinement started from a simple initialization): 12.09 iterations on average, standard deviation 25.32

We recall that the computation time is directly proportional to the number of iterations, and is thus on average three times longer when a random initialization is used instead of Aff-Ref. Regarding the number of iterations, we obtained the results shown in figure 7. We make the following observations. First, the error for Aff-Ref is slightly but consistently lower than the error for Aff-Ref-Rnd and Aff-Ref-Smp. Second, the computation time for Aff-Ref is always lower than for the two other initializations. Depending on the parameter, it can be marginally or substantially lower. For instance, it is quasi-equivalent for the runs with low noise level σ, and 3 orders of magnitude lower for large number of points m.

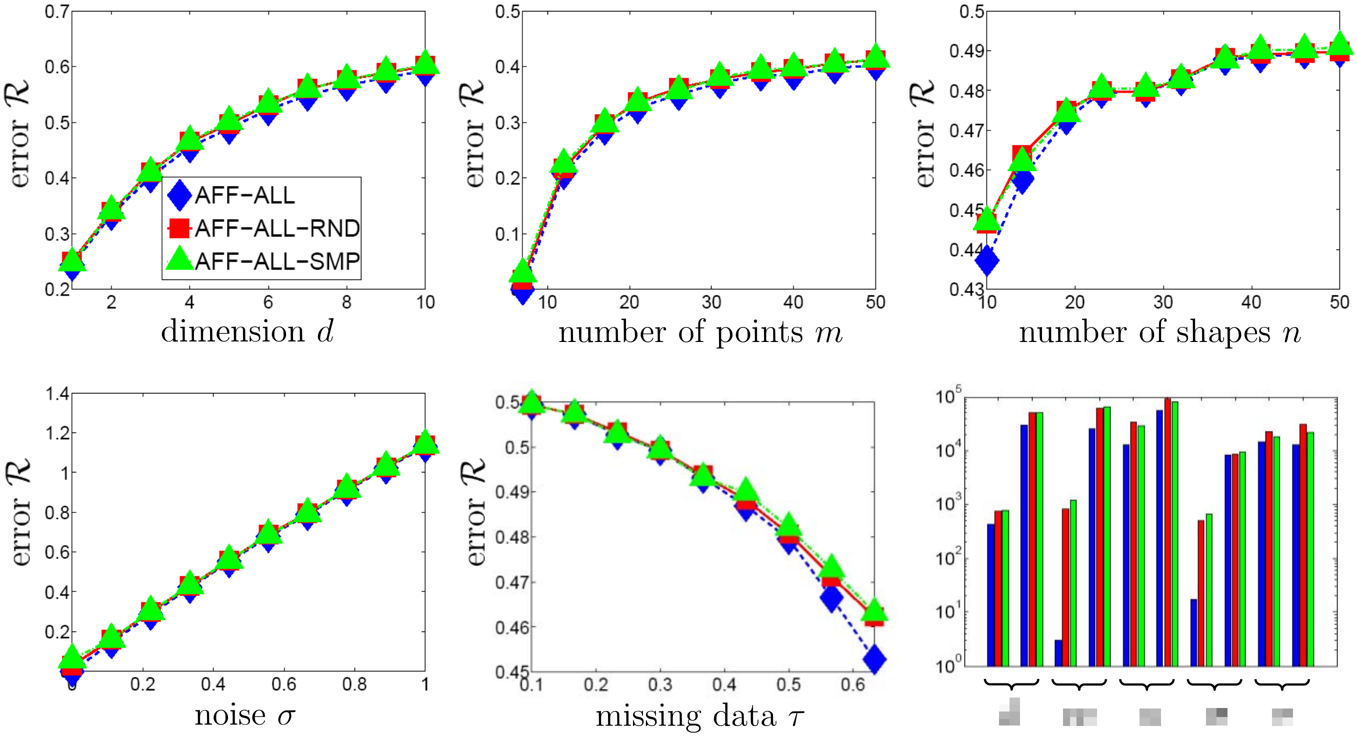


Figure 7: **Influence of the initialization on simulated data.** The first 5 graphs show the error as a function of 5 different parameters, and for 3 different initializations of the nonlinear refinement. The last (bottom-right) graph shows the timing (in ms) for each of the 5 varied parameters and each tested method for the two extreme parameter values: the timing for the minimum and maximum value of the parameters are shown on the left and right batch of bars respectively.

Our conclusions are that the proposed initialization method marginally improves the convergence of the nonlinear refinement, but substantially decreases the number of required iterations and computation time.

7.2.6 Relationship Between the Reference-Space and the Data-Space Models

We investigated the theoretical relationship between the reference-space and data-space models in §2.4. We now present an experiment whose results are shown in figure 8 that illustrates this relationship. In this experiment, the data shapes are randomly generated from a reference shape such that the average scale discrepancy ζ varies from 1 to 2. Four methods are compared. Alternation methods Euc-Alt and Sim-Alt minimize a reference-space error while Euc-All and Sim-All are the proposed methods minimizing a data-space error. As theoretically demonstrated, Euclidean analysis methods Euc-All and Euc-Alt consistently reach the same result. The error depends on the scale discrepancy. None of those methods estimates scale, and the error thus increases with scale discrepancy. On the other hand, similarity analysis methods Sim-All and Sim-Alt consistently reach a different result. Each method compensates for scale discrepancies, and thus reaches an error approximately independent of the scale variations.

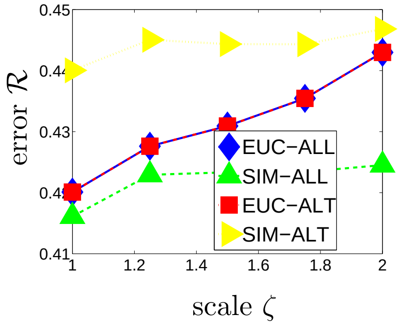


Figure 8: **Relationship between the reference-space and the data-space models on simulated data.** In this experiment the average relative scale ζ between the data shapes is varied from 1 to 2.

7.3 Real Data

We tested the six algorithms described in §7.1 on two real datasets in 2D, 3D and higher dimensions.

7.3.1 The 2D-Face Dataset

The goal of this experiment is to show how our algorithms behave on a real dataset in dimension $d = 2$. The 2D-Face dataset contains $n = 10$ 2D shapes of a face with $m = 40$ points obtained by fitting an Active Appearance Model to a video. It was used in (Bartoli et al., 2008) for nonrigid Structure-from-Motion with the hierarchical low-rank shape model. The tracked face is talking and its shape thus undergoes local deformations. This dataset has missing data with $\tau \approx .1$ since when the face rotates it gets partly self-occluded. Some sample images are shown in figure 9.

We ran different methods for generalized procrustes analysis. Figure 10 shows a graphical rendering of

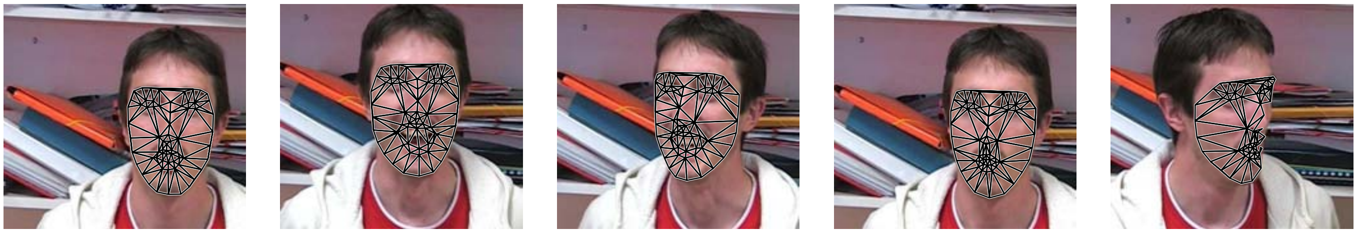


Figure 9: **Sample images from the 2D-Face dataset.** These images were registered using an Active Appearance Model.

the results. The data-space error and timings are given below:

Method	Error (pixels)	Timing (seconds)
Aff-All	6.51	0.53
Sim-All	6.76	1.76
Sim-Alt	6.79	2.03
Euc-All	6.84	1.61
Euc-Alt	6.84	0.07

This confirms our results obtained on simulated data. Aff-All is very fast. Sim-All improves on the classical alternation applied to a similarity: it is both faster and finds a solution with a lower error. Euc-All gives the same error as Euc-Alt but is slower.

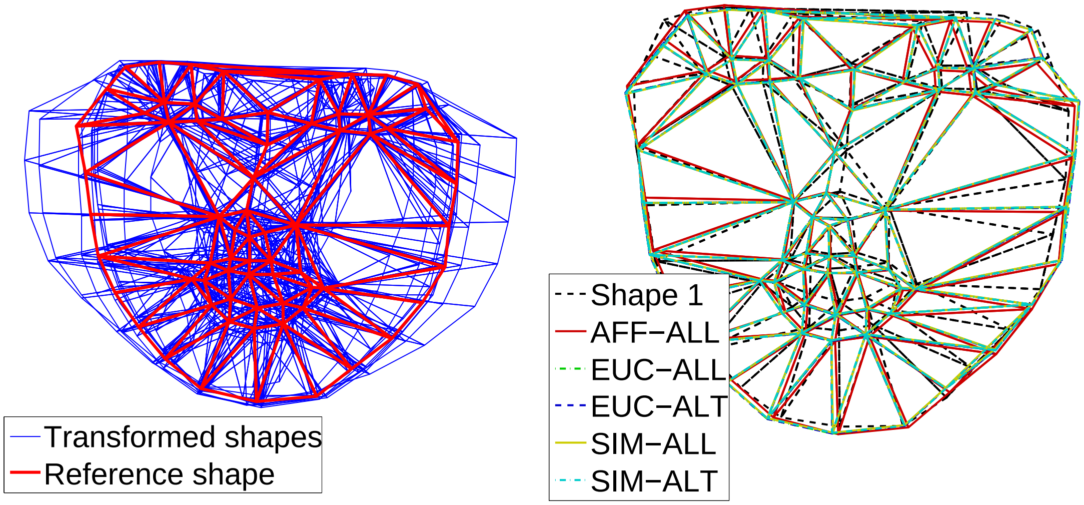


Figure 10: **Results of generalized procrustes analysis on the 2D-Face dataset.** Left: the reference shape and all shape data transformed to the reference coordinate frame for the results of Aff-All. Right: the first shape data with the transformed reference shapes for the 5 tested methods.

7.3.2 The 3D-Human Dataset: Procrustes and Theseus Analysis in Higher Dimensions

The goal of this experiment is to show how our algorithms behave on real datasets in dimension $d = 3$ and higher. The 3D-Human dataset contains $n = 200$ mocap 3D shapes for 5 subjects performing the same action with $m = 20$ points which were extracted from the HumanEVA database (Sigal and Black, 2006). This dataset has no missing data. It shows a walking subject and is thus significantly nonrigid. Some sample shapes are shown in figure 11.

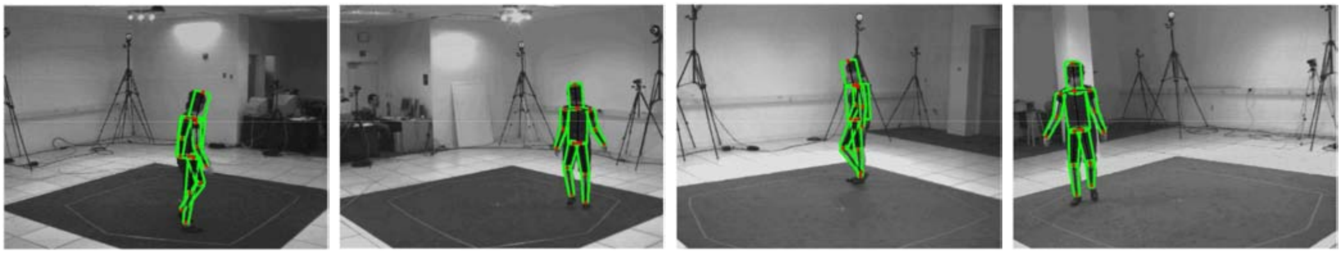


Figure 11: **Sample images from the 3D-Human dataset.** These images were extracted from the HumanEVA database (Sigal and Black, 2006). Motion captured markers are shown as red points.

So as to be able to register the different subjects we take the following route. We will first construct a simple linear deformable model for each subject with l deformation modes and an average shape. We will then register the deformable models with generalized procrustes analysis in dimension $3(l+1)$. We call this second round of generalized procrustes analysis as *generalized theseus analysis*.[7]

Deformable modeling of each subject – 3D generalized procrustes analysis. We do not use a subject's index in this paragraph to keep the notation uncluttered and since the deformable modeling is done independently on each subject. Inspired by the LRSM (Low-Rank Shape Model) which has been extensively used in Non-Rigid Structure-from-Motion (Bregler et al., 2000) we approximate each shape $\mathsf{D}_i^\top \in \mathbb{R}^{3\times m}$ of one subject's walk as:

$$\mathsf{D}_i^\top \;\approx\; \zeta_i \mathsf{E}_i \left(\mathsf{S}^\top + \sum_{k=1}^{l} \varepsilon_{i,k} \mathsf{C}_k^\top \right) + \mathbf{e}_i \mathbf{1}^\top. \tag{32}$$

In this equation ζ_i, E_i and $\mathbf{e}_i$ represent a global similarity transform and S is an average shape. The deformations are modeled by the l configuration shapes coefficients $\varepsilon_{i,k} \in \mathbb{R}$ and the l deformation modes

[7] In Greek mythology Procrustes was a bandit who was stretching people or cutting off their legs so as to make them fit an iron bed's size. Theseus captured Procrustes and fitted him to his own bed.

$\mathsf{C}_k^\top \in \mathbb{R}^{3\times m}$. In Non-Rigid Structure-from-Motion by factorization this equation is rewritten as:

$$\mathsf{D}_i^\top \approx \zeta_i \mathsf{E}_i \mathsf{S}^\top + \begin{pmatrix} \varepsilon_{i,1}\zeta_i \mathsf{E}_i & \cdots & \varepsilon_{i,l}\zeta_i \mathsf{E}_i \end{pmatrix} \begin{pmatrix} \mathsf{C}_1^\top \\ \vdots \\ \mathsf{C}_l^\top \end{pmatrix} + \mathbf{e}_i \mathbf{1}^\top.$$

This shows that the centred measurement matrix is approximately (up to noise and deviation from the model) of rank $3(l+1)$. The factorization is however difficult to perform in practice since the first factor matrix has a block structure which cannot be readily enforced in SVD (or any other matrix factorization.) This introduces additional ambiguities (the number of which is in $\mathcal{O}(9l^2)$) that have to be resolved while trying to enforce the block structure via the mixing matrix (Xiao and Kanade, 2006).

We here take a different route: we first estimate the global motion only using similarity generalized procrustes analysis. We thus first compute S and the ζ_i, E_i and $\mathbf{e}_i$. We then undo the effect of global motion on each shape datum and vectorize the matrix representation to get:

$$\mathbf{d}_i \overset{\text{def}}{=} \operatorname{vect}\left(\frac{1}{\zeta_i} \mathsf{E}_i^{-1} \left(\mathsf{D}_i^\top - \mathbf{e}_i \mathbf{1}^\top \right) - \mathsf{S}^\top \right).$$

This allows us to rewrite the LRSM (32) as:

$$\mathbf{d}_i \approx \sum_{k=1}^{l} \varepsilon_{i,k} \mathbf{c}_k \qquad \text{with} \qquad \mathbf{c}_k \overset{\text{def}}{=} \operatorname{vect}(\mathsf{C}_k^\top).$$

Stacking the n motion compensated and vectorized shape data $\mathbf{d}_i$ in a single measurement matrix gives:

$$\begin{pmatrix} \mathbf{d}_1^\top \\ \vdots \\ \mathbf{d}_n^\top \end{pmatrix}_{(n\times 3m)} \approx \begin{pmatrix} \varepsilon_{1,1} & \cdots & \varepsilon_{1,l} \\ \vdots & \ddots & \vdots \\ \varepsilon_{n,1} & \cdots & \varepsilon_{n,l} \end{pmatrix}_{(n\times l)} \begin{pmatrix} \mathbf{c}_1^\top \\ \vdots \\ \mathbf{c}_l^\top \end{pmatrix}_{(l\times 3m)},$$

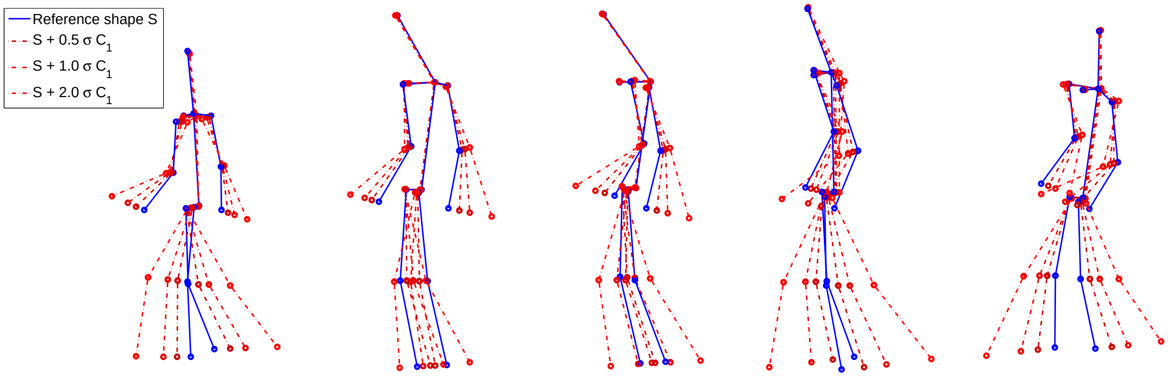
where the index of a matrix indicates its size. This problem can be easily solved by a rank-l factorization of the left-hand side matrix. The ambiguities in this factorization are simple linear reparameterizations and thus do not need to be resolved. In our experiments we kept different numbers of deformation modes: $l = 0, \ldots, 3$.

The following table shows the error obtained for the different similarity algorithms over the 5 different subjects:

Subject	1	2	3	4	5	All
Error (cm) for Sim-Alt	8.61	7.59	7.32	9.77	10.17	8.69
Error (cm) for Sim-Upg	21.26	27.75	16.73	15.74	21.58	20.61
Error (cm) for Sim-All	8.53	7.55	7.29	9.69	10.09	8.63
Error (cm) for the LRSM with $l = 1$	3.59	4.26	3.71	3.75	4.38	3.94
Error (cm) for the LRSM with $l = 2$	2.58	2.13	2.13	2.79	2.82	2.49
Error (cm) for the LRSM with $l = 3$	1.58	1.26	1.26	1.74	1.60	1.49

We observe that the alternation algorithm Sim-Alt gives good results, very close to our nonlinearly refine solution given by Sim-All, while Sim-Upg, our closed-form solution, is able to provide an initial guess which is obviously better than a random guess by still too coarse to be used as is. The LRSM gives lower errors since it models deformations. We observe that the more deformation modes (the higher the value of l), the lower the error, which is perfectly logical.

The average shape and first deformable mode for the different subject are shown in figure 12. We can see that the main variations due to walking motion are well captured. However, we can also clearly see inter-subject differences.



Subject 1 Subject 2 Subject 3 Subject 4 Subject 5

Figure 12: **The Low-Rank Shape Model computed for the 3D-Human dataset.** The average shape and the first mode of variation are shown for the 5 subjects.

Inter-subject deformable registration – generalized theseus analysis in higher dimensions. So far, we obtained an average shape and l deformation modes for each subject. So as to do inter-subject registration we have to compute transforms in both space and time. Indeed, nothing guarantees that the average shape and the deformation modes match between the different subjects. Therefore, we represent each subject's spatio-temporal shape data as a matrix $\mathsf{\Gamma}_p \in \mathbb{R}^{3(l+1)\times m}$ (with $p = 1, \ldots, 5$) defined as:

$$\mathsf{\Gamma}_p^\top \overset{\text{def}}{=} \begin{pmatrix} \mathsf{S}_p & \mathsf{C}_{1,p} & \cdots & \mathsf{C}_{l,p} \end{pmatrix}.$$

We then perform affine generalized theseus analysis by running our affine algorithm in dimension $3(l+1)$. We measure the RMS discrepancy between the original shape data D_i and their prediction from the final computed models, which is proportional to the negative log-likelihood:

Subject	1	2	3	4	5	All
Error (cm) for Aff-Fct with $l = 0$	10.53	7.61	7.34	10.73	11.09	9.46
Error (cm) for Aff-Ref with $l = 0$	12.46	7.90	7.83	10.33	11.34	9.97
Error (cm) for Aff-All with $l = 0$	10.53	7.61	7.34	10.73	11.09	9.46
Error (cm) for Aff-Fct with $l = 1$	4.36	4.41	3.96	4.42	4.86	4.40
Error (cm) for Aff-Ref with $l = 1$	11.66	4.57	4.18	4.13	4.82	5.87
Error (cm) for Aff-All with $l = 1$	4.36	4.41	3.96	4.42	4.86	4.40
Error (cm) for Aff-Fct with $l = 2$	2.94	2.43	2.48	3.10	3.28	2.84
Error (cm) for Aff-Ref with $l = 2$	3.63	2.80	2.66	3.15	3.28	3.11
Error (cm) for Aff-All with $l = 2$	2.94	2.43	2.48	3.10	3.28	2.84
Error (cm) for Aff-Fct with $l = 3$	1.70	1.30	1.33	1.83	1.69	1.57
Error (cm) for Aff-Ref with $l = 3$	2.21	1.58	4.75	3.67	1.88	2.82
Error (cm) for Aff-All with $l = 3$	1.70	1.30	1.33	1.83	1.69	1.57

It is clear that the more deformation modes the lower the error, as expected. Aff-Ref gives remarkably good results, being slightly higher in error than the global minimum of the cost function, always found by Aff-All and Aff-Fct.

8 Conclusion

We presented a novel approach to the generalized procrustes analysis problem that, inspired by similar techniques in Structure-from-Motion, we called the stratified approach. Contrarily to the Structure-from-Motion case, and especially in the affine to similarity and Euclidean upgrade step, we were able for generalized procrustes analysis to give simple methods based on convex linear least squares optimization. Our approach compares favorably with the classical alternation framework: our closest-form solutions are accurate, especially in the important case of Euclidean data. We thus believe that they make an important step in how generalized procrustes analysis can be solved. Nonlinear refinement algorithms are always more accurate than alternation and more widely applicable.

A Demonstrations

A.1 Solution to $\min_{\mathsf{S}} \|\mathsf{LS}\|_{\mathcal{F}}$ such that $\mathsf{S}^\top\mathsf{S} = \mathrm{I}$

Let $\mathsf{L} \in \mathbb{R}^{q\times d}$ we want to show that:

$$\arg\min_{\mathsf{S}\in O(d)} \|\mathsf{LS}\|_{\mathcal{F}}^2 \;=\; \bar{\mathsf{V}},$$

with $\bar{\mathsf{V}}$ the last d columns of matrix V in the SVD $\mathsf{L} \overset{\mathsf{SVD}}{\to} \mathsf{U\Sigma V}^\top$. We write $\mathbf{s}_k$, $k = 1,\ldots,d$ the columns of S (similarly for V with $\mathbf{v}_k$.) We rewrite the above problem as a set of nested problems:

$$\min_{\mathsf{S}\in O(d)} \|\mathsf{LS}\|_{\mathcal{F}}^2 \;=\; \min_{\mathbf{s}_d\in O(d)} \|\mathsf{L}\mathbf{s}_d\|_2^2 + \min_{\mathbf{s}_{d-1}\in O(d),\ \mathbf{s}_d^\top\mathbf{s}_{d-1}=0} \|\mathsf{L}\mathbf{s}_{d-1}\|_2^2 + \cdots$$

The last column $\mathbf{s}_d$ is found first by computing the partial derivatives of the lagrangian:

$$\mathcal{L}_d(\mathbf{s}_d) \;\overset{\mathrm{def}}{=}\; \|\mathsf{L}\mathbf{s}_d\|_2^2 + \lambda(1 - \|\mathbf{s}_d\|_2^2),$$

where λ is a Lagrange multiplier, giving:

$$\mathsf{L}^\top\mathsf{L}\mathbf{s}_d \;=\; \lambda\mathbf{s}_d.$$

This shows that $\mathbf{s}_d$ is a right singular vector of L, and so $\mathbf{s}_d = \mathbf{v}_k$ for some $k \in \{1,\ldots,d\}$. The cost for $\mathbf{v}_k$ is $\|\mathsf{L}\mathbf{v}_k\|_2^2 = \sigma_k^2$ and[8] we therefore choose $\mathbf{s}_d \leftarrow \mathbf{v}_d$ to minimize the cost.

The other columns are then found in turn, as follows. Consider $\mathbf{s}_{d-1}$. The lagrangian is:

$$\mathcal{L}_{d-1}(\mathbf{s}_{d-1}) \;\overset{\mathrm{def}}{=}\; \|\mathsf{L}\mathbf{s}_{d-1}\|_2^2 + \lambda(1 - \|\mathbf{s}_{d-1}\|_2^2) + \mu(\mathbf{s}_{d-1}^\top\mathbf{s}_d)^2,$$

and its partial derivatives are:

$$\frac{1}{2}\frac{\partial\mathcal{L}_{d-1}}{\partial\mathbf{s}_{d-1}} \;=\; \mathsf{L}^\top\mathsf{L}\mathbf{s}_{d-1} - \lambda\mathbf{s}_{d-1} + \mu\mathbf{s}_d^\top\mathbf{s}_d\mathbf{s}_{d-1} \;=\; 0.$$

Since $\mathbf{s}_{d-1}^\top\mathbf{s}_{d-1} = \|\mathbf{s}_{d-1}\|_2^2 = 1$ this gives:

$$\mathsf{L}^\top\mathsf{L}\mathbf{s}_{d-1} \;=\; (\lambda - \mu)\mathbf{s}_{d-1}.$$

[8] We have that $\|\mathsf{L}\mathbf{v}_k\|_2^2 = \|\mathsf{U\Sigma V}^\top\mathbf{v}_j\|_2^2 = \|\mathsf{U\Sigma}\mathbf{e}_k\|_2^2 = \sigma_k^2\|\mathbf{u}k\|_2^2 = \sigma_k^2$ where $\mathbf{e}_k$ is a zero vector with one as its k-th component.

Similarly to $\mathbf{s}_d$ we finally find that $\mathbf{s}_{d-1} \leftarrow \mathbf{v}_k$ for some $k \in \{1, \ldots, d\}$, and choose $\mathbf{s}_{d-1} \leftarrow \mathbf{v}_{d-1}$ to minimize the cost and satisfy the constraints. Doing the same reasoning for all the remaining columns leads to $\mathsf{S} \leftarrow \bar{\mathsf{V}}$.

A.2 Translational Gauge Constraints for the Reference-Space Cost

The reference-space cost is translation invariant, which means that we can simply fix the translation part of the gauge by adding a gauge constraint to the cost, as if it were a simple penalty. We chose to centre the reference shape using $\mathsf{S}^\top \mathbf{1} = \mathbf{0}$. Below, we prove that the original cost function is invariant to translations of the reference shape. Note that a translation represented by vector $\mathbf{g} \in \mathbb{R}^d$ applies to the reference shape S as $\mathsf{S} \;\; \rightarrow \;\; \mathsf{S} + \mathbf{1}\mathbf{g}^\top$ with $\mathbf{1} \in \mathbb{R}^m$.

Consider the minimization problem (15), and plug in the translational gauge transformation of the reference shape mentioned directly above:

$$\|(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathsf{S}\|_{\mathcal{F}}^2 \;\; \rightarrow \;\; \|(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S(\mathsf{S} + \mathbf{1}\mathbf{g}^\top)\|_{\mathcal{F}}^2.$$

This is expanded to $\|(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathsf{S} + (\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathbf{1}\mathbf{g}^\top\|_{\mathcal{F}}^2$. Consider the second factor. By construction, matrix $\mathsf{K}_S \in \mathbb{R}^{\alpha \times m}$ is a stack of identity matrices to which some rows were removed, and thus $\mathsf{K}_S\mathbf{1} = \mathbf{1} \in \mathbb{R}^\alpha$, leading to:

$$(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathbf{1}\mathbf{g}^\top \;\; = \;\; \mathbf{1}\mathbf{g}^\top - \hat{\mathsf{K}}_B\mathbf{1}\mathbf{g}^\top. \tag{33}$$

We have that $\hat{\mathsf{K}}_B = \operatorname{diag}\left(\hat{\mathsf{K}}_{B,i}\right)$ and $\mathsf{K}_{B,i} = (\mathsf{D}_i \;\; \mathbf{1})$. Denoting $\mathbf{d}_i = \mathsf{D}_i^\top \mathbf{1}$, we get:

$$\hat{\mathsf{K}}_{B,i} \;\; = \;\; \mathsf{K}_{B,i}\left(\mathsf{K}_{B,i}^\top \mathsf{K}_{B,i}\right)^{-1}\mathsf{K}_{B,i}^\top \;\; = \;\; \begin{pmatrix} \mathsf{D}_i & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathsf{D}_i^\top \mathsf{D}_i & \mathbf{d}_i \\ \mathbf{d}_i^\top & m \end{pmatrix}^{-1} \begin{pmatrix} \mathsf{D}_i^\top \\ \mathbf{1}^\top \end{pmatrix},$$

from which, defining $\mathsf{H}_i \overset{\text{def}}{=} \left(\mathsf{D}_i^\top \mathsf{D}_i - \frac{1}{m}\mathbf{d}_i\mathbf{d}_i^\top\right)^{-1}$ we get:

$$\hat{\mathsf{K}}_{B,i}\mathbf{1} \;\; = \;\; \begin{pmatrix} \mathsf{D}_i & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathsf{H}_i & -\frac{1}{m}\mathsf{H}_i\mathbf{d}_i \\ -\frac{1}{m}\mathbf{d}_i^\top \mathsf{H}_i^\top & \frac{1}{m}\left(1 + \frac{1}{m}\mathbf{d}_i^\top \mathsf{H}_i \mathbf{d}_i\right) \end{pmatrix} \begin{pmatrix} \mathbf{d}_i \\ m \end{pmatrix} \;\; = \;\; \begin{pmatrix} \mathsf{D}_i & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \;\; = \;\; \mathbf{1}.$$

This shows from equation (33) that $(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathbf{1}\mathbf{g}^\top = \mathbf{0}$ and that the reference-space cost is thus translational gauge invariant: $\|(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S\mathsf{S}\|_{\mathcal{F}}^2 = \|(\mathrm{I} - \hat{\mathsf{K}}_B)\mathsf{K}_S(\mathsf{S} + \mathbf{1}\mathbf{g}^\top)\|_{\mathcal{F}}^2$.

A.3 The Closest Special Orthonormal Matrix

We give an simple demonstration to find the closest special orthonormal $\mathsf{E} \in SO(d)$ matrix to a matrix M with $\det(\mathsf{M}) > 0$ mimizing the cost $\mathcal{C} \overset{\text{def}}{=} \|\mathsf{E} - \mathsf{M}\|_{\mathcal{F}}^2$. Different solutions and proofs exist in the literature ((Horn et al., 1988; Schönemann and Carroll, 1970)); they are more involved than ours. We first plug the SVD $\mathsf{M} \overset{\mathsf{SVD}}{\rightarrow} \mathsf{U}\Sigma\mathsf{V}^\top$ in the cost, and since $\mathsf{U} \in O(d)$ and $\mathsf{V} \in O(d)$ we get $\mathcal{C} = \|\mathsf{H} - \Sigma\|_{\mathcal{F}}^2$ with $\mathsf{H} \overset{\text{def}}{=} \mathsf{U}^\top\mathsf{E}\mathsf{V}$ and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$. We are left with the problem of finding the closest orthonormal matrix to a diagonal matrix. Expanding the norm in the cost, we get $\mathcal{C} = \|\mathsf{H}\|_{\mathcal{F}}^2 + \|\Sigma\|_{\mathcal{F}}^2 - 2\text{vect}(\mathsf{H})^\top\text{vect}(\Sigma)$ that we rewrite as:

$$\mathcal{C} \;=\; d + \sigma_1 + \cdots + \sigma_d - 2(H_{1,1}\sigma_1 + \cdots + H_{d,d}\sigma_d).$$

Given that $\mathsf{H} \in O(d)$ we have that $H_{k,k} \leq 1$, $k = 1, \dots, d$. Noting that $\sigma_k \geq 0$, $k = 1, \dots, d$, the solution that minimizes the expression of $\mathcal{C}$ directly above is given by choosing $H_{k,k} = 1$, $k = 1, \dots, d$, leading to $\mathsf{H} = \mathrm{I}$. We finally get the solution:

$$\mathsf{E} \;=\; \mathsf{U}\mathsf{V}^\top.$$

It is to be noted that $\sigma_k > 0$, $k = 1, \dots, d$ then $\det(\Sigma) > 0$. Since $\det(\mathsf{M}) > 0$, $\mathsf{U} \in O(d)$ and $\mathsf{V} \in O(d)$ this implies that $\det(\mathsf{E}) = \det(\mathsf{U})\det(\mathsf{V}) = 1$ and thus $\mathsf{E} \in SO(d)$ as sought.

We now look into the demonstration of the scaled case, where we have to find both matrix $\mathsf{E} \in SO(d)$ and a scale factor $\zeta > 0$ such that the following cost $\mathcal{S} \overset{\text{def}}{=} \|\zeta\mathsf{E} - \mathsf{M}\|_{\mathcal{F}}^2$ is minimized. As in the above case, we use the SVD of M to get $\mathcal{S} = \|\zeta\mathsf{H} - \Sigma\|_{\mathcal{F}}^2$ where H is defined as above. We expand the norm in the cost as above and get $\mathcal{S} = \|\zeta\mathsf{H}\|_{\mathcal{F}}^2 + \|\Sigma\|_{\mathcal{F}}^2 - 2\text{vect}(\zeta\mathsf{H})^\top\text{vect}(\Sigma)$ that we rewrite as:

$$\mathcal{S} \;=\; d\zeta^2 + \sigma_1 + \cdots + \sigma_d - 2\zeta(H_{1,1}\sigma_1 + \cdots + H_{d,d}\sigma_d).$$

The same argument as above directly leads to $\mathsf{H} = \mathrm{I}$ (and thus $\mathsf{E} = \mathsf{U}\mathsf{V}^\top$). This leaves us with:

$$\mathcal{S} \;=\; d\zeta^2 + \sigma_1 + \cdots + \sigma_d - 2\zeta(\sigma_1 + \cdots + \sigma_d).$$

Setting the derivatives of $\mathcal{S}$ with respect to ζ to as to find the minimizer gives:

$$\frac{1}{2}\frac{\partial\mathcal{S}}{\partial\zeta} \;=\; d\zeta - (\sigma_1 + \cdots + \sigma_d) \;=\; 0,$$

and readily leads to:

$$\zeta \;=\; \frac{1}{d}\sum_{k=1}^{d}\sigma_k.$$

It is easily verified that $\zeta > 0$ as required.

B Implementation: the Generalized Procrustes Analysis Toolbox

We created a Matlab toolbox that implements our proposed algorithms. We called it the *generalized procrustes analysis toolbox*; it will be released under the GPL licence. Our toolbox offers a ready-to-use interface for registering shape data using any of the described methods for affine, Euclidean and similarity transformations. It also includes the classical alternation approach for similarity and Euclidean transformations and copes with incomplete shape data. **Our toolbox is submitted as supplemental material – all our experiments can thus be replicated very easily (see below).**

B.1 Main Function

The whole set of algoritms is bundled in a single function called `gpa`. It has the following syntax:

```
T=gpa(D,V,options);
```

where `D` contains the shape data, `V` indicates the missing points and `options` specifies which algorithm to use, the level of verbosity and the maximum number of iterations. The output `T` is a structure that contains the computed transformations, the error for the different trials and the computational time for various steps of the algorithms. Of course, a detailed help can be obtained by typing:

```
help gpa;
```

B.2 Simulating Data and Testing

All the experiments with simulated data that we reported in this paper can be replicated using an experimental front-end function. It has a compact interface that generates data and then calls the `gpa` function. This function is called `genExperiment` and has the following syntax:

```
genExperiment(options,name);
```

There is no output since the results will be saved in a file. `options` is a structure that defines the method or methods that will be tested, the number of shape data, the dimension, the ratio of missing data, the amount of added noise or of deformation between the shapes, the parameter that will be varied, the number of random trials, the name of the file to save the results and the type of data (transformations) to be generated. The second input parameter called `name` specifies the name of a file containing the results of an experiment which is yet to be completed and from which the current experiment will be started.

As an example of usage of `genExperiment`, we give the piece of code that replicates experiment the top-row noise graph shown in figure 4:

```
options.title='exp1RMSvssigma';
options.n=5;
options.d=3;
options.m=50;
options.tau=0;
options.sigma=sqrt(0.01);
options.varpar='sigma';
options.parvalues=linspace(0,sqrt(0.1),10);
options.iterations=100;
options.methods={'AFF-FCT','AFF-REF','AFF-ALL'};
options.transformations='AFF'; % or 'EUC' or 'SIM'
genExperiment(options);
```

The results will be saved in a file called `exp1RMSvssigma.mat`. If the experiment could not complete for some reason it can be restarted from where it stopped by typing:

```
genExperiment(options,'exp1RMSvssigma');
```

Of course, a detailed help for this function can be obtained by typing:

```
help genExperiment;
```

References

P. Aguiar, J. Xavier, and M. Stosic. Globally optimal solution to exploit rigidity when recovering structure from motion under occlusion. In *International Conference on Image Processing*, 2008.

K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D points sets. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

A. Bartoli, D. Pizarro, and M. Loog. Stratified generalized procrustes analysis. In *British Machine Vision Conference*, 2010.

P. T. Boggs, J. R. Donaldson, and R. B. Schnabel. ODRPACK: Software for weighted orthogonal distance regression. ACM *Transactions on Mathematical Software*, 15:348–364, 1989.

C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *International Conference on Computer Vision and Pattern Recognition*, 2000.

I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* John Wiley and Sons, 1998.

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*, 9:272–290, 1997.

D. Goryn and S. Hein. On the estimation of rigid body rotation from noisy data. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1219–1220, December 1995.

J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

J. C. Gower and G. B. Dijksterhuis. *Procrustes Problems.* New York: Oxford University Press, 2004.

P. Gurdjos, A. Bartoli, and P. Sturm. Is dual linear self-calibration artificially ambiguous? In *International Conference on Computer Vision*, 2009.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2003. Second Edition.

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, April 1987.

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):1127–1135, July 1988.

K. Kanatani. Analysis of 3-D rotation fitting. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–549, May 1994.

S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian. Global registration of multiple 3D point sets via optimization-on-a-manifold. In *Eurographics Symposium on Geometry Processing*, 2005.

S. Mahamud, M. Herbert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure and motion. In *International Conference on Computer Vision and Pattern Recognition*, 2001.

B. Matei and P. Meer. Optimal rigid motion estimation and performance evaluation with bootstrap. In *International Conference on Computer Vision and Pattern Recognition*, 1999.

N. Ohta and K. Kanatani. Optimal estimation of three-dimensional rotation and reliability evaluation. In *European Conference on Computer Vision*, 1998.

M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

L. Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–105, May 1996.

J. A. Ramos and E. I. Verriest. Total least squares fitting of two point sets in m-D. In *Conference on Decision and Control*, 1997.

P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.

P. H. Schönemann and R. M. Carroll. On fitting one matrix to another under choice of a central dilation transformation and a rigid motion. *Psychometrika*, 35(2), June 1970.

L. Sigal and M. J. Black. HumanEva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

C. C. Slama, editor. *Manual of Photogrammetry, Fourth Edition*. American Society of Photogrammetry and Remote Sensing, 1980.

J. M. F. Ten Berge. Orthogonal procrustes rotation for two or more matrices. *Psychometrika*, 42(2):267–276, 1977.

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon. Bundle ajustment — a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, 2000.

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

M. W. Walker, L. Shao, and R. A. Volz. Estimating 3D location parameters using dual number quaternions. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54(3):358–367, November 1991.

G. Wen, D. Zhu, S. Xia, and Z. Wang. Total least squares fitting of point sets in m-D. In *Computer Graphics International*, 2005.

J. Xiao and T. Kanade. A linear closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, March 2006.

J. Xiao, B. Georgescu, X. Zhou, D. Comaniciu, and T. Kanade. Simultaneous registration and modeling of deformable shapes. In *International Conference on Computer Vision and Pattern Recognition*, 2006.

A. J. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *International Journal of Computer Vision*, 53(2):153–167, March 2003.