Global Optimization for Optimal Generalized Procrustes Analysis

Daniel Pizarro
University of Alcala
Alcala de Henares, Spain
pizarro@depeca.uah.es

Adrien Bartoli
Université d'Auvergne
Clermont-Ferrand, France
adrien.bartoli@gmail.com

Abstract

This paper deals with generalized procrustes analysis. *This is the problem of registering a set of shape data by estimating a reference shape and a set of rigid transformations given point correspondences. The transformed shape data must align with the reference shape as best possible. This is a difficult problem. The classical approach computes alternatively the reference shape, usually as the average of the transformed shapes, and each transformation in turn.*

We propose a global approach to generalized procrustes analysis for two- and three-dimensional shapes. It uses modern convex optimization based on the theory of Sum Of Squares functions. We show how to convert the whole procrustes problem, including missing data, into a semidefinite program. Our approach is statistically grounded: it finds the maximum likelihood estimate.

We provide results on synthetic and real datasets. Compared to classical alternation our algorithm obtains lower errors. The discrepancy is very high when similarities are estimated or when the shape data have significant deformations.

1. Introduction

This paper deals with the problem of rigid registration between different input shapes represented by point correspondences. This is known as *procrustes analysis* in the statistics and shape analysis literature [2, 4]. More precisely, it is called *generalized procrustes analysis* when more than two shapes are to be registered. In this problem, one transformation per observed shape has to be computed. The shape data are mapped to a *reference shape* which is as well to be estimated. Let d be the dimension of the shape data to be analyzed. The estimated transformations are similarities (scaling, rotation and translation) with $\frac{1}{2}d(d+1)+1$ degrees of freedom (dof) or euclidean transformations (without scaling and $\frac{1}{2}d(d+1)$ dof.) Figure 1 illustrates the generalized procrustes analysis problem.

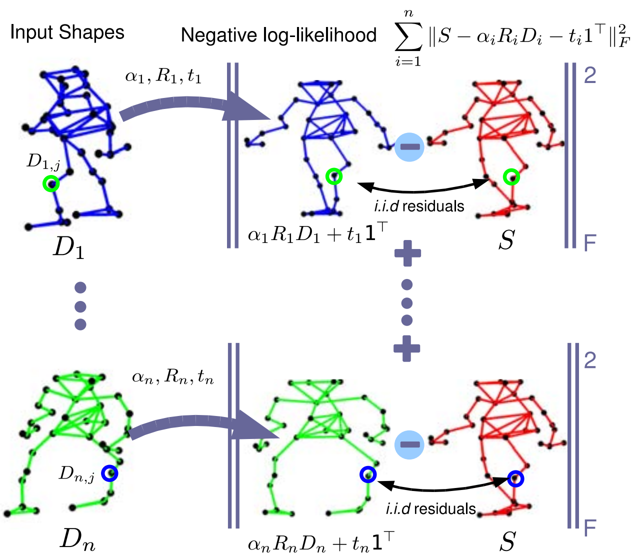


Figure 1. The generalized procrustes analysis problem is, given n input shape data (in blue and green), to compute a reference shape S (in red) and n similarity or euclidean transformations (one per input shape.) Minimizing the sum of squares of the discrepancies between the registered input shapes and the reference shape points yields the maximum likelihood estimate. Our proposed algorithms find the global solution to this optimal generalized procrustes analysis statement in 2D and 3D, while the literature only provides iterative local optimization methods such as the popular alternation [2, 4, 6, 15] (to cite just a few) and the recent stratification approach [1].

The classical approach to generalized procrustes analysis [2, 4, 6, 15] selects one of the shape data as a reference shape and registers each of the other shapes to the reference in turn by solving the absolute orientation problem [3, 5, 14]. It then alternates 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. The alternation approach is iterative and does not guarantee convergence to the global minimum of the cost function. In [1] a stratified approach is proposed where procrustes analysis is first solved with affine transformations. The solution is then 'upgraded' to similarity or euclidean transformations. The main advantage of the stratified approach is that, unlike alternation, it obtains all transformations simultaneously. It has better performance than the alternation approach but still uses iterative Newton based optimization. It is thus not guaranteed to find the global minimum of the cost function either.

Our paper uses the optimal cost function (in the sense of maximum likelihood) that involves all sought transformations and the unknown reference shape. We propose an algorithm that, unlike the alternation and the stratified approaches, always finds the global minimum. It is based on the recent Sum Of Squares (SOS) theory [7, 10] from algebraic geometry that allows one to find global bounds of polynomials with both equality and inequality constraints [9].

We first show how to express generalized procrustes analysis for similarity and euclidean transformations in 2D and 3D as a polynomial optimization problem, with polynomial constraints. Based on the SOS theory, we then show how to obtain the global minimum of the cost using an SOS Program (SOSP) relaxation of the original problem [7], and its translation into a convex semidefinite program (SDP) [13].

This paper is organized as follows. The problem statement, state of the art on procrustes analysis and insights into the SOS theory are given in §2. Our global and optimal approach to generalized procrustes analysis is presented in §3. Experimental results on simulated and real data are reported in §4 and conclusions drawn in §5. Finally, appendix A gives details on the SOS theory and its use in optimization.

2. Preliminaries and Previous Work

2.1. Problem Statement

We define $T \triangleq \{R_1, t_1, \alpha_1, \ldots, R_n, t_n, \alpha_n\}$ to be the set of n similarities $T_i : \mathbb{R}^d \rightarrow \mathbb{R}^d$ where $R_i \in SO(d)$ are rotation matrices, $t_i \in \mathbb{R}^d$ are translation vectors and $\alpha_i > 0$ are scale factors. Euclidean transformations are obtained as a special case by setting $\alpha_1 = \cdots = \alpha_n = 1$. The input shapes are represented by n matrices $D_1, \ldots, D_n$. Each shape $D_i \in \mathbb{R}^{d \times m}$ is composed of m d-dimensional points:

$$D_i = \begin{pmatrix} D_{i,1} \cdots D_{i,m} \end{pmatrix} \qquad D_{i,j} \in \mathbb{R}^d. \tag{1}$$

The problem to be solved can be cast as the one of finding the set of similarities T and the reference shape $S = (S_1 \cdots S_m) \in \mathbb{R}^{d \times m}$ that minimize the cost function $\mathcal{E}$ defined as:

$$\mathcal{E}(T, S) = \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j} \left\| S_j - \alpha_i R_i D_{i,j} - t_i \right\|_2^2, \tag{2}$$

where $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top \mathbf{u}}$ is the vector two-norm. The variables $v_{i,j} \in \{0, 1\}$ allow us to model *missing data*, the fact that some points may not be observed in some shapes. The cost function (2) is not gauge invariant. Some constraints are thus needed to fix the reference frame. This cost is also called reference-space model; it corresponds to the negative log-likelihood for *i.i.d.* gaussian noise (see [1] for more details.) Equation (2) can also be written as:

$$\mathcal{E}(T, S) = \sum_{i=1}^{n} \left\| (S - \alpha_i R_i D_i - t_i \mathbf{1}^\top) V_i \right\|_F^2, \tag{3}$$

where $\|\mathbf{u}\|_F = \sqrt{\operatorname{tr}(\mathbf{u}^\top \mathbf{u})}$ is the Frobenius norm, $\mathbf{1} \in \mathbb{R}^{m \times 1}$ is an all-one vector and $V_i \in \mathbb{R}^{m \times m}$ is the diagonal matrix $V_i = \operatorname{diag}(v_{i,1}, \cdots, v_{i,m})$. The final problem statement is obtained by adding the constraints that the transformations in T are similarities:

$$\min_{T,S} \mathcal{E}(T, S) \quad R_i \in SO(d) \quad \alpha_i > 0 \quad i = 1, \cdots, n. \tag{4}$$

2.2. Previous Work

Most procrustes analysis algorithms in the literature are based on the idea of alternating the estimation of the transformations and of the reference shape. One possible implementation is:

1. Initialize the transformations to identity ($R_i = I,\ \alpha_i = 1,\ t_i = 0;\ \forall i$)
2. Compute the reference shape as the average of registered shapes:
$$S = \sum_{i=1}^{n} \left(\alpha_i R_i D_i + t_i \mathbf{1}^\top \right) \left(\sum_{i=1}^{n} V_i \right)^{-1}$$
3. Compute each transformation in turn by solving the absolute orientation problem (see below): for $i = 1, \cdots, n$, compute T_i from D_i, S and V_i
4. Stop if the reference shape does not change else goto 2

Most of the implementations of the alternation approach differ by how they solve the absolute orientation problem (both iterative [3] and algebraic closed-form solutions [5, 14] were proposed) and how they bootstrap the algorithm (for instance, one may replace step 1 by setting the reference shape to one of the shape data.) Integrated alternation solutions for multiple shapes using total least squares have been recently proposed [6, 15].

A recent approach based on the concept of stratification used in Structure-from-Motion has been proposed in [1]. Instead of alternating, the reference shape and the set of transformations are computed simultaneously using a convex cost function that approximates the negative log likelihood. Affine transformations are computed first; they are then upgraded to euclidean or similarity transformations.

The result is global but suboptimal, and is finally refined using Newton like minimization.

To summarize, the literature does not have any global optimization solution to the optimal (maximum likelihood) formulation of generalized procrustes analysis. We propose such an approach, that deals with both similarity and euclidean transformations in 2D and 3D, and reduces the problem to a simple SDP, by using the recent SOS theory.

2.3. Optimization and Sum of Squares Theory

We briefly describe how to convert a constrained polynomial optimization problem into an SOSP. More details on the SOS theory [7, 10] and its use in optimization [9, 11] can be found in the literature.

Many optimization problems are to find bounds of multivariate polynomials under polynomial constraints:

$$\begin{aligned} &\min f(x) && && (5)\\ &\text{subject to} && g_i(x) \geq 0, \quad i = 1, \cdots, M \\ & && h_j(x) = 0, \quad j = 1, \cdots, N \end{aligned}$$

This is an NP-Hard and non-convex problem for which a global solution cannot be generally found. However, some modern results in the field of algebraic geometry [7, 10] have shown that one can find a computational relaxation method that 'converts' a polynomial optimization problem into an SDP in some cases. This is based on the SOS theory; a polynomial $f(x)$ is an SOS if $f(x) = \sum_{i=1}^{n} f_i^2(x)$. The SOS condition is stricter than non-negativity and generally more computationally tractable.

The SOS relaxation approach to solve the optimization problem (5) first replaces inequalities with SOS conditions and includes an scalar variable γ that represents the lower bound of the polynomial $f(x)$. Second, it uses auxiliary variables (polynomials $\lambda_i(x)$ and SOS polynomials $\sigma_j(x)$) in order to include all constraints in the cost. This technique is based on the so-called *positivstellensatz* property, a central result in algebraic geometry, that converts the constrained optimization problem (5)into the following general SOSP:

$$\begin{aligned} &\min && -\gamma && (6)\\ &\text{subject to} && f(x) - \gamma - (\sigma_0(x) + \textstyle\sum_j \lambda_j(x) h_j(x) + \\ & && \textstyle\sum_i \sigma_i(x) g_i(x) + \dots) \text{ is an SOS} \\ & && \sigma_i(x) \text{ is an SOS} \quad i = 1, \cdots, M \end{aligned}$$

Every SOSP can be exactly converted to an equivalent semidefinite program (SDP) and thus powerful convex optimization tools such as SeDuMi [13] can be used to globally solve any SOSP. [7, 8] give a comprehensive explanation about the exact relationship between an SOSP and an SDP. In fact, there are tools [9] available that solve any SOSP using SDP solvers and do the conversion automatically.

The solution of the SOSP gives a lower bound γ of the polynomial and the value of x for that bound. As it is commented in [8], for some "rare" examples of polynomials, the bound obtained can be useless and thus the SOS relaxation is not near enough to the original problem.

3. Global Generalized Procrustes Analysis

We describe next our approach to the generalized procrustes analysis problem. It is optimal in the sense of maximum likelihood: our approach is guaranteed to find a global minimum of the negative log-likelihood function; it is based on the SOS theory. We first show how the optimal unknown translations and reference shape can be computed in closed-form. The closed-forms are then substituted in the original cost function.

We then examine the gauge properties of the reduced cost function and show how to find the optimal rotations and scale factors. Our approach is applicable in 2D and in 3D, the two cases of interest in shape analysis, and can be used for euclidean and similarity transformations.

It is worth of note that at no time in the derivation below we approximate the negative log likelihood: our algorithms end up minimizing it exactly and globally in all cases.

3.1. The Translations and Reference Shape

We show how to convert the problem presented in equation (4) into a reduced one, where the set of unknowns consists of scale factors $\alpha_1, \cdots, \alpha_n$ and rotation matrices $R_1, \cdots, R_n$.

The translations. The optimal translations $t_1, \cdots, t_n$ are first obtained from the cost function (3). By setting $\frac{\partial \mathcal{E}}{\partial t_i}(T, S) = 0$ we obtain:

$$t_i = \frac{1}{m_i} \sum_{j=1}^{m} (S_j - \alpha_i R_i D_{i,j}) v_{i,j} = \frac{1}{m_i} (S - \alpha_i R_i D_i) V_i \mathbf{1}, \quad (7)$$

where $m_i = \sum_{j=1}^{m} v_{i,j}$ is the number of visible points in shape D_i. By introducing equation (7) into (3) the following reduced cost is obtained:

$$\mathcal{E}(\bar{T}, S) = \sum_{i=1}^{n} \left\| (S - \alpha_i R_i D_i) \left(I - V_i \frac{\mathbf{1}\mathbf{1}^\top}{m_i} \right) V_i \right\|_F^2, \quad (8)$$

where $\bar{T} = \{R_1, \alpha_1, \cdots, R_n, \alpha_n\}$, and I is the identity matrix.

The reference shape. By differentiating the reduced cost (8) as $\frac{\partial \mathcal{E}}{\partial S}(\bar{T}, S) = 0$ we obtain:

$$SM = \sum_{i=1}^{n} \alpha_i R_i D_i M_i, \quad (9)$$

with $M_i = \left(I - \frac{V_i \mathbf{1}\mathbf{1}^\top}{m_i}\right) V_i$ and $M = \sum_{i=1}^n M_i$. Matrix M must thus be inverted to compute S. It is an $m \times m$ symmetric matrix having the following properties:

- Matrix M has, at least, a single zero eigenvalue. The vector $\mathbf{1}^\top$ is the basis of the left nullspace of M (*i.e.* $\mathbf{1}^\top M = 0^\top$). The proof is straightforward; since $\mathbf{1}^\top V_i \mathbf{1} = m_i$:

$$M_i \mathbf{1} = \mathbf{1}^\top - \mathbf{1}^\top \frac{V_i \mathbf{1}\mathbf{1}^\top}{m_i} = 0;\ \forall i. \tag{10}$$

- If a point j is missing in all shapes (*i.e.* $v_{i,j} = 1 \quad \forall i$), the dimension of the left nullspace of M is larger than one. The proof is also straightforward; since in the case of a missing point j in all shapes, all matrices V_i share the zero j-th row and column, and so does M. In this case any all-zero vector with a non-zero constant in its j-th position belongs to the left nullspace of M.

Assuming that all points are seen by at least one shape, the nullspace of M accounts for the fact that the cost function is invariant to a constant offset in $S \rightarrow S + t_o \mathbf{1}^\top$ and $t_i \rightarrow t_i + t_o$. An extra constraint is needed to solve equation (9) for an unique S, fixing t_o to a specific value. We propose to force the reference shape's centre of gravity to lie at the origin:

$$S\mathbf{1} = \sum_{j=1}^m S_j = 0. \tag{11}$$

This means that $SV_i\mathbf{1} + S(I - V_i)\mathbf{1} = 0;\ \forall i$ and:

$$SV_i\mathbf{1} = -S(I - V_i)\mathbf{1};\ \forall i. \tag{12}$$

Taking into account equation (12), if $S\mathbf{1} = 0$ the following equality holds:

$$SM = S \sum_{i=1}^n \left(I + \frac{(I - V_i)\mathbf{1}\mathbf{1}^\top}{m_i}\right) V_i = S\tilde{M}, \tag{13}$$

Contrarily to matrix M, matrix $\tilde{M}$ is invertible if all shape points are seen at least by one of the shapes. By introducing equation (13) in (9), which implicitly imposes the constraint $S\mathbf{1} = 0$, we can thus get S as:

$$S = \sum_{i=1}^n \left(\alpha_i R_i \hat{D}_i\right) \tilde{M}^{-1}, \tag{14}$$

where $\hat{D}_i = D_i \left(I - \frac{V_i \mathbf{1}\mathbf{1}^\top}{m_i}\right) V_i$. By substituting S from equation (14) in equation (8) a new reduced cost function is obtained in terms of rotation matrices and scale factors:

$$\mathcal{E}(\bar{T}) = \sum_{i=1}^n \left\| \left(\sum_{k=1}^n \alpha_k R_k \hat{D}_k\right) \tilde{M}^{-1} \left(I - \frac{V_i \mathbf{1}\mathbf{1}^\top}{m_i}\right) V_i - \alpha_i R_i \hat{D}_i \right\|_F^2. \tag{15}$$

For simplicity equation (15) is rewritten:

$$\mathcal{E}(\bar{T}) = \sum_{i=1}^n \left\| \sum_{k=1}^n \alpha_k R_k \tilde{D}_k^i - \alpha_i R_i \hat{D}_i \right\|_F^2, \tag{16}$$

where $\tilde{D}_k^i = \hat{D}_k \tilde{M}^{-1} \left(I - \frac{V_i \mathbf{1}\mathbf{1}^\top}{m_i}\right) V_i$.

The main difficulty in obtaining the global minimum of (16) comes from the $SO(d)$ constraints for each matrix R_i. Such constraints are nonlinear and nonconvex in the coefficients of R_i and there is no linear parameterization that could enforce it. Assuming an iterative and non-linear optimization strategy (*e.g.* Sequential Quadratic Programming) as a possible solution, the constraints can be iteratively imposed but it is not guaranteed that a global minimum is reached, even if a proper initialization were provided.

3.2. Gauge Freedoms and Degeneracy

It is clear from equation (16) that the reduced cost function is invariant to a rotation R_o applied to the whole set of rotation matrices $R_1, \ldots, R_n$ (the set $R_o R_1, \ldots, R_o R_n$ is equivalent.) For unicity we fix $R_1 = I$.

Unlike the rotation, the cost function is not invariant to a scale factor applied to the set of scales $\alpha_1, \ldots, \alpha_n$: the problem has a trivial solution with $\mathcal{E} = 0$ if $\alpha_1 = \cdots = \alpha_n = 0$. To remove this degeneracy we set $\alpha_1 = 1$.

3.3. Rotations and Scales in 3D

Principle. The minimization of equation (16) for $d = 3$ can be rewritten as a multivariate minimization problem in $5n$ variables (where n is the number of shapes.) First, for each rotation matrix R_i its equivalent unitary quaternion is used $q_i = \begin{pmatrix} q_{i,1} & q_{i,2} & q_{i,3} & q_{i,4} \end{pmatrix}$. Each rotation matrix R_i is then expressed as a vector r_i of polynomials in q_i:

$$r_i = \begin{pmatrix} q_{i,1}^2 + q_{i,2}^2 - q_{i,3}^2 - q_{i,4}^2 \\ 2q_{i,2}q_{i,3} + 2q_{i,1}q_{i,4} \\ 2q_{i,2}q_{i,4} - 2q_{i,1}q_{i,3} \\ 2q_{i,2}q_{i,3} - 2q_{i,1}q_{i,4} \\ q_{i,1}^2 + q_{i,3}^2 - q_{i,2}^2 - q_{i,4}^2 \\ 2q_{i,3}q_{i,4} + 2q_{i,2}q_{i,1} \\ 2q_{i,2}q_{i,4} + 2q_{i,1}q_{i,3} \\ 2q_{i,3}q_{i,4} - 2q_{i,1}q_{i,2} \\ q_{i,1}^2 + q_{i,4}^2 - q_{i,2}^2 - q_{i,3}^2 \end{pmatrix}. \tag{17}$$

Equation (16) can be expressed as:

$$\mathcal{E}(\bar{T}) = \sum_{i=1}^n \sum_{j=1}^m \left\| \left(\sum_{k=1}^n \alpha_k R_k \tilde{D}_{k,j}^i - \alpha_i R_i \hat{D}_{i,j} \right) \right\|^2. \tag{18}$$

By using the quaternion parameterization we get:

$$\mathcal{E}(\bar{T}) = \sum_{i=1}^n \sum_{j=1}^m \left\| \frac{1}{n} \sum_{k=1}^n \tilde{A}_{k,j}^i \alpha_k r_k - \hat{A}_{i,j} \alpha_i r_i \right\|^2, \tag{19}$$

where:

$$\tilde{A}^i_{k,j} = \begin{pmatrix} \left(\tilde{D}^i_{k,j}\right)^\top & 0_{1\times 3} & 0_{1\times 3} \\ 0_{1\times 3} & \left(\tilde{D}^i_{k,j}\right)^\top & 0_{1\times 3} \\ 0_{1\times 3} & 0_{1\times 3} & \left(\tilde{D}^i_{k,j}\right)^\top \end{pmatrix},$$

and:

$$\hat{A}_{i,j} = \begin{pmatrix} \hat{D}^\top_{i,j} & 0_{1\times 3} & 0_{1\times 3} \\ 0_{1\times 3} & \hat{D}^\top_{i,j} & 0_{1\times 3} \\ 0_{1\times 3} & 0_{1\times 3} & \hat{D}^\top_{i,j} \end{pmatrix},$$

By naming $\bar{r}$ the following $9n$ vector:

$$\bar{r} = \begin{pmatrix} \alpha_1 r_1 \\ \vdots \\ \alpha_n r_n \end{pmatrix}, \tag{20}$$

equation (19) is rewritten as:

$$\mathcal{E}(\bar{T}) = \bar{r}^\top \left(\sum_{j=1}^{m} A_j^\top A_j \right) \bar{r} = \bar{r}^\top L \bar{r}, \tag{21}$$

where $A_j \in \mathbb{R}^{3n\times 9n}$:

$$A_j = \begin{pmatrix} \hat{A}_{1,j} - \tilde{A}^1_{1,j} & -\tilde{A}^1_{2,j} & \cdots & -\tilde{A}^1_{n,j} \\ -\tilde{A}^2_{1,j} & \hat{A}_{2,j} - \tilde{A}^2_{2,j} & \cdots & -\tilde{A}^2_{n,j} \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{A}^n_{1,j} & -\tilde{A}^n_{2,j} & \cdots & \hat{A}_{n,j} - \tilde{A}^n_{n,j} \end{pmatrix}.$$

The uniqueness and non-degeneracy constraints $R_1 = I_{3\times 3}$ and $\alpha_1 = 1$ could be imposed directly in the SOSP as regular equalities. However it is more efficient to modify equation (21) and remove the first quaternion and scale. Let $\tilde{r}$ be defined as:

$$\bar{r}^\top = \left(1, 0, 0, 0, 1, 0, 0, 0, 0, 1, \tilde{r}^\top\right), \tag{22}$$

the cost function becomes:

$$\mathcal{E}(\bar{T}) = \tilde{r}^\top \tilde{L} \tilde{r} + L_c \tilde{r} + L_{cc}, \tag{23}$$

where matrices $\tilde{L}$, L_c and L_{cc} are easily derived from matrix L. It is important to note that the size of L, and consequently of $\tilde{L}$, L_c and L_{cc} is not dependent on the number of points m but on the number n of shapes. Moreover, the structure of matrix L allows us to compute it with complexity linear in m and so this step is at worst $\mathcal{O}(n^2 m)$. Since matrix L has a fixed size independent of m it implies that the resulting SDP has a fixed complexity.

The cost function (23) is a multivariate polynomial in the quaternion coefficients and scales. Its bound can be thus obtained using an SOSP relaxation.

Rotations only. So as to compute euclidean transformations we set the scales to $\alpha_1 = \cdots = \alpha_n = 1$. Therefore, for each quaternion to represent a valid $SO(3)$ matrix, its norm must be one ($\|q_i\| = 1$) and $q_{i,1} \geq 0$ in order to resolve the ambiguities (*i.e.* q_i and $-q_i$ represent the same rotation.) Problem (16) can finally be written as:

$$\begin{aligned} \min \quad & \tilde{r}^\top \tilde{L} \tilde{r} + L_c \tilde{r} + L_{cc} \\ \text{subject to} \quad & q_{i,1} \geq 0 \quad i = 2, \cdots, n \\ & \|q_i\|^2 = 1 \quad i = 2, \cdots, n \end{aligned} \tag{24}$$

The cost function (24) can be minimized as an SOSP with the *positivstellensatz* formulation.

Rotations and scales. For similarities, we include the scales and simply impose the constraints $\alpha_i > 0$ for all scales. This can be done in an efficient way by noticing that each vector r_i has terms composed of second order monomials in the entries of q_i It easily follows that $\alpha_i r_i = \hat{r}_i$, where $\hat{r}_i$ is given by the non-unitary quaternion:

$$\hat{q}_i = \sqrt{\alpha_i} q_i. \tag{25}$$

Using non-unitary quaternions the problem becomes:

$$\begin{aligned} \min \quad & \tilde{r}^\top \tilde{L} \tilde{r} + L_c \tilde{r} + L_{cc} \\ \text{subject to} \quad & \hat{q}_{i,1} \geq 0 \quad i = 2, \cdots, n, \end{aligned} \tag{26}$$

After optimization the scales are obtained as $\alpha_1 = 1$ and $\alpha_i = \|\hat{q}_i\|$ for $i = 2, \ldots, n$.

3.4. Rotations and Scales in 2D

Principle. The cost function (16) for $d = 2$ is simpler than in 3D as in 2D rotation matrices can be easily parameterized by unitary vectors $q_i = (q_{i,1}, q_{i,2})$:

$$r_i = \begin{pmatrix} q_{i,1} & q_{i,2} & -q_{i,2} & q_{i,1} \end{pmatrix}^\top. \tag{27}$$

Equation (19) remains similar to the 3D case using:

$$\tilde{A}^i_{k,j} = \begin{pmatrix} \left(\tilde{D}^i_{k,j}\right)^\top & 0_{1\times 2} \\ 0_{1\times 2} & \left(\tilde{D}^i_{k,j}\right)^\top \end{pmatrix} \text{ and } \hat{A}_{i,j} = \begin{pmatrix} \hat{D}^\top_{i,j} & 0_{1\times 2} \\ 0_{1\times 2} & \hat{D}^\top_{i,j} \end{pmatrix}.$$

The vector $\bar{r}$ has now size $2n \times 1$ and is identical to (20). The optimization problem is equally cast as the quadratic equation (23) in vector $\tilde{r}$, resulting from removing the first scale and rotation from vector $\bar{r}$.

Rotations only. The euclidean version of the problem includes unitary constraints:

$$\begin{aligned} \min \quad & \tilde{r}^\top \tilde{L} \tilde{r} + L_c \tilde{r} + L_{cc} \\ \text{subject to} \quad & ||q_i||^2 = 1 \quad i = 2, \cdots, n \end{aligned} \tag{28}$$

Rotations and scales. In the similarity version we again use non-unitary vectors $\hat{q}_i = \alpha_i q_i$ giving the following unconstrained problem:

$$\min \quad \tilde{r}^\top \tilde{L} \tilde{r} + L_c \tilde{r} + L_{cc} \tag{29}$$

The scales are obtained after optimization as in the 3D case.

4. Experimental Results

We tested our algorithms with both simulated and real data. In all cases we tested the euclidean (EUC-SOS) and the similarity (SIM-SOS) versions of our algorithms in 2D and 3D. We compared our algorithmed to classical alternation (EUC-ALT and SIM-ALT for euclidean and similarity respectively.) We used the alternation algorithm drafted in §2.2 using the solution of [5] to the absolute orientation problem. As in our algorithm we fix $R_1 = I$ and $\alpha_1 = 1$ in alternation algorithms. In all experiments we compared the value of the cost function (3) between the different algorithms; this reflects the accuracy that is reached since this cost is the negative log likelihood. SOSP are solved using SeDuMi and the SOSTools package [9].

4.1. Simulated Data

Data generation. The reference shape is generated by drawing $m = 50$ points in an origin-centered hyper-sphere of unit radius. Each of the $n = 5$ affine transformations is randomly generated by selecting $d+1$ control points into the unit hypersphere. Euclidean transformations are obtained as the orthonormal part of each affine transformation using QR factorization and ensuring positiveness of the determinant. The generated shapes are obtained by applying the n transformations to the m reference points. Non-rigidity and noise are both modeled by an additive gaussian process with variance $\sigma^2 = 1$. Missing data are obtained by erasing points with $\tau = 0.5$ (50% missing data.)

Experiments. Each experiment is run 100 times and we report average (RMS) values. The ranges are the following: $\sigma^2 = 0 \ldots 2$; $n = 2, \ldots, 5$; $m = 10, \ldots, 50$; $d = 2, 3$; $\tau = 0 \ldots 7$.

As can be seen in figure 2 both EUC-ALT and EUC-SOS achieve the same level of accuracy in all experiments. Surprisingly the alternation approach is performing very well for euclidean transformations. This is however not the case when scales are incorporated to the problem. In those cases the alternation approach gets stuck in a local minimum in almost all experiments while SIM-SOS reduces the error considerably.

4.2. Real Data

We tried our algorithms on two real datasets. The first dataset contains $n = 5$ 2D shapes with $m = 40$ 2D points that represent a human face with different poses in front of a camera. This dataset includes missing data ($\tau \approx 0.1$) when the face turns, as it suffers from self-occlusion.

In figure 3a to 3d the reference shape is shown in thick in front of the set of transformed shapes $\alpha_i R_i D_i + t_i$ for all methods. This gives an idea of the error in the reference frame. Below each figure the error is given. As was expected SIM-SOS outperforms the other methods. The computational times of this experiment are shown below figure 3. The proposed methods are slower than alternation.

The second dataset corresponds to 3D point coordinates given by Motion Capture (MOCAP) sensors, available from the HumanEVA [12] database. The dataset has $n = 5$ shapes corresponding to a walking person. As in the face dataset, we show in figure 3e to 3h the reference shape and the set of transformed shapes for each method.

The same conclusions are obtained from this experiment; SIM-SOS achieves the lowest error. The processing time in this case increases compared to the 2D case.

5. Conclusions

We have proposed a global solution to the optimal (maximum likelihood) generalized procrustes analysis problem. Our algorithms are the first in the literature that feature these characteristics. Experimental results show that the popular alternation approach falls in local minima when similarities are estimated, while the global minimum is at a significantly lower cost, as our algorithms show. Our algorithms use the SOS theory and reduce the problem at hand to simple SDP, then solved using SeDuMi.

One possible improvement on which we are currently working concerns the computational time required by our algorithms. They are more computationally expensive than simple alternation, taking seconds to register a couple of shapes in our Matlab implementation.

References

[1] A. Bartoli, D. Pizarro, and M. Loog. Stratified Generalized Procrustes Analysis. In *British Machine and Vision Conference*, 2010.

[2] I. Dryden and K. Mardia. *Statistical shape analysis*. Wiley New York, 1998.

[3] D. Eggert, A. Lorusso, and R. Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5):272–290, 1997.

[4] J. Gower and G. Dijksterhuis. *Procrustes problems*. Oxford University Press, USA, 2004.

[5] B. Horn et al. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[6] S. Krishnan, P. Lee, J. Moore, and S. Venkatasubramanian. Global registration of multiple 3D point sets via

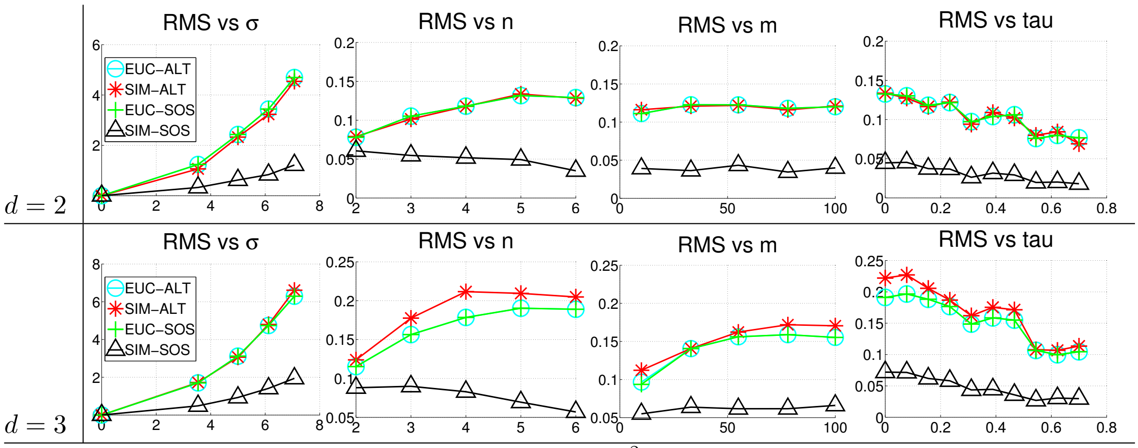
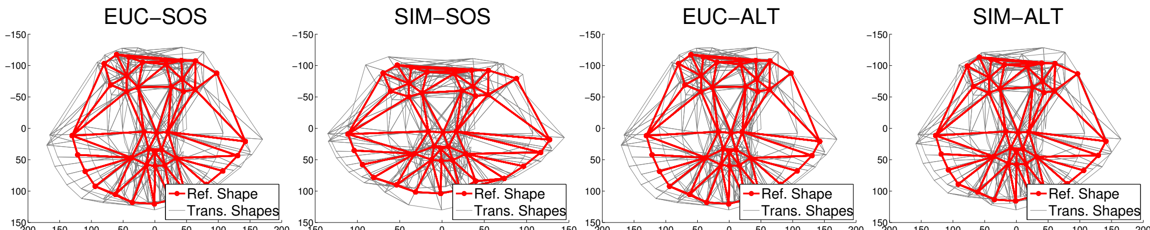
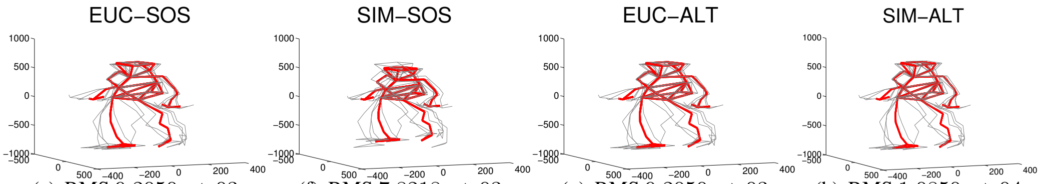


Figure 2. Experimental results Experiment 1: varying σ^2, n, m, d and τ (see main text for details.)



(a) RMS 293.1708 (b) RMS 222.5163 (c) RMS 293.1708 (d) RMS 262.7479

Algorithm	EUC-SOS	SIM-SOS	EUC-ALT	SIM-ALT
Computation time (seconds)	1.74	0.21	0.09 (14 iterations)	0.19 (28 iterations)



(e) RMS $9.2050e+03$ (f) RMS $7.8218e+03$ (g) RMS $9.2050e+03$ (h) RMS $1.0850e+04$

Algorithm	EUC-SOS	SIM-SOS	EUC-ALT	SIM-ALT
Computation time (seconds)	142	28	0.12 (14 iterations)	0.23 (30 iterations)

Figure 3. Experimental setup with 3D and 2D real datasets (see main text for details.)

optimization-on-a-manifold. In *Eurographics Symposium on Geometry Processing*, 2005.

[7] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[8] P. Parrilo and B. Sturmfels. Minimizing polynomial functions. *American Mathematical Society*, 60:83–99, 2001.

[9] S. Prajna, A. Papachristodoulou, and P. Parrilo. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *IEEE conference on decision and control*, 2001.

[10] K. Schmudgen. The K-moment problem for compact semialgebraic sets. *Mathematische Annalen*, 289(1):203–206, 1991.

[11] G. Schweighofer and A. Pinz. Globally optimal $O(n)$ solution to the PnP problem for general camera models. In *British Machine and Vision Conference*, 2008.

[12] L. Sigal and M. Black. HumanEVA: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 2006.

[13] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1):625–653, 1999.

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

[15] G. Wen, D. Zhu, S. Xia, and Z. Wang. Total least squares fitting of point sets in mD. In *Computer Graphics International*, pages 82–86, 2005.