Determinant-Constrained Affine Transformations

Adrien Bartoli[1,3] Daniel Pizarro[2,3]

[1] Department of Clinical Research and Innovation
Clermont-Ferrand University Hospital

[2] Departamento de Electrónica
Universidad de Alcalá

[3] EnCoV, IGT, Institut Pascal
UMR6602 CNRS / Université Clermont Auvergne

Corresponding author: Adrien Bartoli
Adrien.Bartoli@gmail.com

Abstract

We study the problem of estimating an affine transformation with prescribed determinant from point pairs in dimension d. The determinant constraint is nonconvex and challenging to handle. We propose a method to reformulate the problem into a system of d quadratics and 1 polynomial of degree d in $d+1$ unknowns. We propose a solution method for this system and a procedure to retrieve the single or multiple affine transformation solutions. We show that the related problem of finding the closest d-simplex of prescribed volume and orientation to a given arbitrary d-simplex is solved as a special case of the proposed method. We report experimental results demonstrating the use of the proposed transformations in three applications.

Code and data release. We have publicly shared our implementation for all the proposed computation procedures and experiments, including the required data to ensure reproducibility, accessible from the authors' personal or GitHub pages.

Contents

1 Introduction

We study the least-squares estimation of an affine transformation with prescribed determinant, given point pairs in $\mathbb{R}^d$, where d is the dimension, with $d \geq 2$. We name such a transformation a Determinant-Constrained Affine Transformation (DCAT) and its estimation as the DCAT problem. The affine transformation has a linear part represented by a matrix $A \in \mathbb{R}^{d\times d}$ and a translation part represented by a vector $t \in \mathbb{R}^d$. The m point-pair coordinates are held in matrices $P, Q \in \mathbb{R}^{d\times m}$ where column j of P matches column j of Q. We prescribe the determinant of A to be equal to $s \in \mathbb{R}_*$, where $\mathbb{R}_* = \mathbb{R} \setminus \{0\}$. The DCAT problem is thus formulated as:

$$\min_{A,t} \left\| \left(AP + te^\top - Q \right) W \right\|^2 \quad \text{s.t.} \quad \det(A) = s, \tag{1}$$

where $e \in \mathbb{R}^m$ is the all-one vector, $W \in \mathbb{R}^{m\times m}$ is a given diagonal non-zero-weight matrix, and $\| \cdot \|$ is the matrix Frobenius norm. We assume that the point pairs are in general position and so that they determine a unique general affine transformation. Hence, we have $m \geq d + 1$ and $\mathrm{rk}([P^\top e]) = \mathrm{rk}([Q^\top e]) = d + 1$. The cost in the DCAT problem is convex but the constraint involves the determinant, which is a polynomial expression of degree d, making the problem nonconvex and non trivial. The use of general-purpose optimisation techniques such as gradient descent and Newton-Raphson would thus not guarantee finding an optimal solution. In contrast, our main contribution is a procedure to solve the DCAT problem (1) exactly. Specifically, our solution procedure finds the constrained global minimum of the cost function numerically. It explicitly finds all the local minima and selects the global one, without using iterative minimisation. It may also compute all solutions, should there be multiple ones caused by ambiguities.

Our main motivation to study the DCAT problem is that the absolute value of the determinant controls the rate of volume change and the sign of the determinant controls the orientation change. Specifically, defining the volume of a subset $S \subset \mathbb{R}^d$ as $\mathcal{V}(S)$ and the transformed subset by the affine transformation (A, t) as $\mathcal{T}(S; A, t)$, we have:

$$\mathcal{V}(\mathcal{T}(S; A, t)) = |\det(A)|\, \mathcal{V}(S). \tag{2}$$

Consequently, any combination of volume and orientation constraints may be formulated as $\det(A) = s$ in equation (1): solving the DCAT problem allows one to control volume and orientation changes. DCAT can thus estimate affine transformations whose linear part belongs to the special linear group $SL_d(\mathbb{R})$, which is the Lie group of volume- and orientation-preserving linear transformations, and its extension $SL_d^{\pm}(\mathbb{R})$, which is the Lie group of volume-preserving linear transformations. Estimating a transformation in several other groups and sets is well-understood, including the orthogonal group $O_d = \{M \in \mathbb{R}^{d\times d}, M^\top M = \mathrm{I}\}$ representing roto-reflections, the special orthogonal group SO_d representing rotations, the group of matrices representing similarities (Umeyama, 1991) and the general set $\mathbb{R}^{d\times d}$ representing affine transformations (Hartley and Zisserman, 2003). Estimating a transformation in the general linear group $GL_d(\mathbb{R})$ is not possible because it represents the non-singular transformations characterised by a non-zero determinant, forming an open set. Indeed, the general point-pair positions guarantee that the affine transformation is unique but not that it is full rank, which we discuss in comment 2. Estimating a guaranteed non-singular transformation in $SL_d(\mathbb{R})$ and $SL_d^{\pm}(\mathbb{R})$ with DCAT may thus be used in 'replacement' of $GL_d(\mathbb{R})$ in some application cases. In particular, the $SL_2(\mathbb{R})$ group contains the so-called squeeze mappings, which are equivalent to hyperbolic rotations and form a basis for the Lorentz boost, from which the transformations of special relativity are constructed (Taylor and Wheeler, 1966).

In addition to DCAT, we study the related problem of least-squares estimation of the closest d-simplex with prescribed volume and orientation to a given d-simplex. We name this problem the Prescribed Volume and Orientation Simplex (PVOS). The given and closest d-simplices are each represented by $d+1$ points held in matrices $M, N \in \mathbb{R}^{d\times(d+1)}$. We assume that the given points in M are in general position, hence $\det([M^\top\, e]) \neq 0$. The point order defines the simplex orientation. The simplex volume and orientation are given by the signed volume $\mathcal{W}(N)$, which we prescribe to be equal to a given $h \in \mathbb{R}_*$. The PVOS problem is thus formulated as:

$$\min_N \|(N - M)W\|^2 \quad \text{s.t.} \quad \mathcal{W}(N) = h, \tag{3}$$

where $W \in \mathbb{R}^{(d+1)\times(d+1)}$ is a given diagonal non-zero-weight matrix. The solution to problem (3) is also the orthogonal projector on the $d(d+1)-1$ dimensional manifold of simplices of some prescribed volume and orientation. While the existing literature does not provide any solution to the DCAT problem, the PVOS problem was recently solved for the specific dimension $d = 2$ and $W = \mathrm{I}$ (Arango and Bartoli, 2021). This solution follows a lengthy derivation and requires a complex solution procedure. Importantly, it does not extend to dimensions higher than $d = 2$. In contrast, we show that the general PVOS problem can be formulated as a specific DCAT problem and solved as such in any dimension $d \geq 2$.

We explore the application of DCAT and PVOS in Laparoscopic Augmented Reality from Preoperative data (LARP) in liver surgery (Adagolodjo et al., 2017; Koo et al., 2017) based on position-based dynamics methods (Bouaziz et al., 2014; Kee et al., 2021). We explore two other applications given as appendix. One is approximate local volume- and orientation-preserving interactive image warping with Moving Least Squares (MLS) (Schaefer et al., 2006), which uses DCAT with weights prescribed by the MLS principle. The other one is Generalised Procrustes Analysis (GPA) (Gower, 1975), which uses DCAT through a standard alternation scheme for multiple data shapes.

2 Determinant-Constrained Affine Transformation

We start with a theoretical analysis of the DCAT problem, from which we show how the problem can be expressed in a standard formulation. We then give a solution to the standard formulation. Finally, we give a solution procedure which implements the proposed method in pseudo-code.

2.1 Standardisation

We start with a standardisation of the problem, where we cancel the translation and absorb the weights in the point coordinates. This then allows us to analyse the rank of the source point coordinates.

We follow the standard translation elimination method by data centring, as for instance in affine structure-from-motion (Tomasi and Kanade, 1992). Starting from the DCAT problem formulation (1), we form the Lagrangian of the problem with the Lagrange multiplier $\lambda \in \mathbb{R}$ as:

$$\mathcal{L} = \frac{1}{2}\left\|\left(AP + te^\top - Q\right)W\right\|^2 + \lambda(\det(A) - s), \tag{4}$$

which we expand as:

$$\mathcal{L} = \frac{1}{2}\left(\left\|te^\top W\right\|^2 + 2\operatorname{tr}\left(WP^\top A^\top te^\top W\right) - 2\operatorname{tr}\left(WQ^\top te^\top W\right)\right) + \epsilon, \tag{5}$$

where ϵ groups the remaining terms independent of t. In particular, as the constraint is independent of the translation, it is thus incorporated in ϵ. We rearrange the terms as:

$$\mathcal{L} = \frac{1}{2}\left(\|W\|^2\|t\|^2 + 2t^\top APW^2e - 2t^\top QW^2e\right) + \epsilon. \tag{6}$$

Upon differentiating, we obtain:

$$\|W\|^2 t + APW^2e - QW^2e = 0, \tag{7}$$

and the solution:

$$t = \frac{1}{\|W\|^2}(Q - AP)W^2e = \bar{q} - A\bar{p}, \tag{8}$$

where $\bar{q} = \frac{1}{\|W\|^2}QW^2e$ and $\bar{p} = \frac{1}{\|W\|^2}PW^2e$ are the weighted centroids of the point sets. Finally, substituting the translation in the original problem and expanding, we obtain the *standard problem formulation*:

$$\min_A \left\|AP' - Q'\right\|^2 \quad \text{s.t.} \quad \det(A) = s, \tag{9}$$

expressed from the centred weighted point sets $P' = \left(P - \bar{p}e^\top\right)W$ and $Q' = \left(Q - \bar{q}e^\top\right)W$.

2.2 Reduction

We reduce the problem to an equivalent problem with d virtual point pairs. We use an RQ factorisation $P' = [R\,0]B^\top$, where $R \in \mathbb{R}^{d\times d}$ is an upper-triangular matrix with $\operatorname{rk}(R) = d$, $0 \in \mathbb{R}^{d\times(m-d)}$ is an all-zero matrix and $B \in O_m$ is an orthonormal matrix. Substituting the factorisation into the standard formulation (9), we have:

$$\min_A \left\|A[R\,0]B^\top - Q'\right\|^2 \quad \text{s.t.} \quad \det(A) = s, \tag{10}$$

which, from the orthonormality of B, can be rewritten as:

$$\min_A \left\|A[R\,0] - Q'B\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{11}$$

Introducing the partition $B = [B_1\,B_2]$, with $B_1 \in \mathbb{R}^{m\times d}$ and $B_2 \in \mathbb{R}^{m\times(m-d)}$ and expanding, we have:

$$\min_A \left\|AR - Q'B_1\right\|^2 + \left\|Q'B_2\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{12}$$

The second term is independent of the free variable A and can thus be ignored. By defining the d virtual point pairs as $P'' = R$ and $Q'' = Q'B_1$, with $P'', Q'' \in \mathbb{R}^{d\times d}$, we arrive at the *reduced standard problem formulation*:

$$\min_A \left\|AP'' - Q''\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{13}$$

Comment 1 (RQ factorisation ambiguities)**.** *For a full-rank landscape matrix, the RQ factorisation has ambiguities of two types (Björck, 1996, §1.3.1). The first ambiguity is $R \to RS$ and $B_1 \to B_1S$ for $S = \operatorname{diag}(\pm1,\ldots,\pm1) \in O_d$. The second ambiguity is $B_2 \to B_2Z$ for $Z \in O_{m-d}$. Both ambiguities leave the cost function and all matrix characteristics invariant, and, as a change of variable was not used, do not change the solution and can thus be ignored.*

We define the *joint rank* as $r = \operatorname{rk}(Q'')$. As $Q'' = Q'B_1$ and B_1 is a basis for the range of P', we have $r = \operatorname{rk}(Q'B_1) = \operatorname{rk}(Q'P'^\top) = \operatorname{rk}(P'Q'^\top)$.

Comment 2 (Joint-rank deficiency)**.** *As $Q'' = Q'B_1$ and $\mathrm{rk}(Q') = \mathrm{rk}(B_1) = d$, the joint-rank r is generally full. However, we formally have $r \leq \min(\mathrm{rk}(Q'), \mathrm{rk}(B_1)) = d$, hence cases exist where $r < d$. As we required the points to be in general position, these cases are not due to special configurations of a point set such as coplanarity, but they are due to special cases of coordinate arrangements. Such cases can be generated numerically. Figure 1 shows two examples of centred paired point sets P', Q' of $m = 10$ points in dimension $d = 2$. In the first example, $r = 1$, with $\mathrm{sv}_1(Q'B_1) = 0.3087$ and $\mathrm{sv}_2(Q'B_1) = 5.7242e - 18$, where sv_i is the i-th singular value. In the second example, $r = 0$, with $\mathrm{sv}_1(Q'B_1) = 1.5101e - 16$ and $\mathrm{sv}_2(Q'B_1) = 1.2754e - 17$.*

We define *proper point pairs* as point pairs $P, Q \in \mathbb{R}^{d\times m}$, $m > d$, for which the joint rank r is full, meaning $r = d$. In other words, proper point pairs lead to a non-singular unconstrained affine transformation, whilst *improper point pairs* lead to a singular one. Though improper point pairs can thus be an issue in unconstrained affine transformation estimation, they are not for the DCAT problem which, thanks to the determinant constraint, guarantees that the estimated transformation is always non-singular.

Lemma 1 (Minimal problems and joint-rank deficiency)**.** *Minimal problems correspond to $m = d + 1$ point pairs. A minimal problem with each point set $P, Q \in \mathbb{R}^{d\times d+1}$ in general position is proper. Hence, joint-rank deficiency may only occur for $m > d + 1$.*

Proof. We follow the standardisation of section 2.1. We define $g = W^{-1}e \in \mathbb{R}^{d+1}$, which is a right nullvector of P' and Q'. Therefore, the rows of P' being orthogonal to g, they cannot be in the kernel of Q', from which we arrive at $r = \mathrm{rk}(Q'P'^{\top}) = d$. □

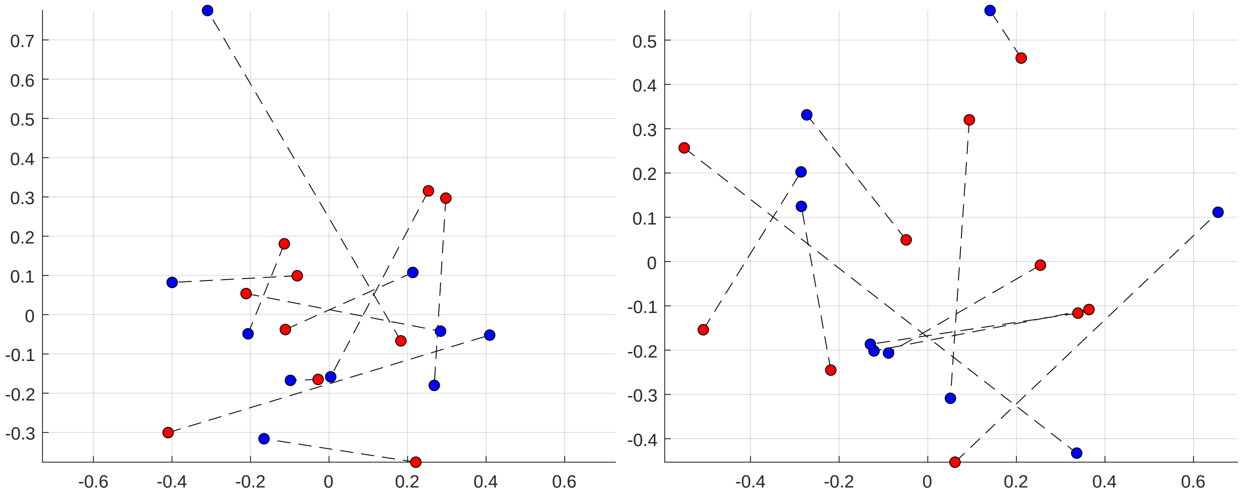


Figure 1: Examples of joint-rank r deficiency. The 10 point pairs are shown as red-blue connected dots. Each point set is in general position. (left) Example with $r = 1$. (right) Example with $r = 0$.

2.3 Orthogonalisation

We perform a step which we name orthogonalisation, which gains independence of the coordinate frames where the virtual point pairs P'' and Q'' are expressed. We achieve this by using the invariance of the cost function to orthonormal transformations. We use an SVD $Q'' = U\Gamma V^{\top}$, where $U, V \in O_d$ are orthonormal matrices and $\Gamma \in \mathbb{R}^{d\times d}$ is a diagonal matrix. We introduce this decomposition in the reduced standard formulation (13), pre-multiply by $U^{\top}$ and post-multiply by V, giving:

$$\min_{A} \left\| U^{\top} A P'' V - \Gamma \right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{14}$$

Upon changing variable by introducing $A' = U^\top A P'' V$, $P''' = \mathrm{I}$ and $Q''' = \Gamma$, we arrive at the *orthogonal standard problem formulation*:

$$\min_{A'} \left\|A' - \Gamma\right\|^2 \quad \text{s.t.} \quad \det(A') = s', \tag{15}$$

where $s' = \det(A') = \det(U^\top A P'' V) = s \det(R) \det(U) \det(V)$. We give a solution method for A' from the orthogonal formulation in the next section. Once A' is solved, we obtain the sought solution for A by inverting the change of variable as:

$$A = U A' V^\top P''^{-1}. \tag{16}$$

Comment 3 (SVD decomposition ambiguities)**.** *The singular values are unique, whereas the singular vectors are not (Björck, 1996, §1.2.1). In particular, for a full-rank matrix, a sign-flipping ambiguity of corresponding singular vectors occurs for simple (non repeated) singular values and a symmetric orthogonal mixing of singular vectors occurs for multiple (repeated) singular values. In addition, for a rank-deficient matrix, an asymmetric orthogonal mixing of singular vectors occurs for vanishing singular values. Owing to the change of variable from A to A', some of these ambiguities impact the solution when using the reverse change of variable (16). These ambiguities are modelled by $U \to U K_u$ and $V \to V K_v$, $K_u = \operatorname{diag}(Z, Z_u)$, $K_v = \operatorname{diag}(Z, Z_v)$, where Z, Z_u, Z_v are orthonormal matrices. Specifically, the leading block $Z \in O_r$ models sign flipping and symmetric orthogonal mixing. It is block-diagonal with orthonormal blocks, whose sizes match the numbers of repeated singular values $b_1, \dots, b_l$ as $Z = \operatorname{diag}(Z_1, \dots, Z_l)$ with $Z_k \in O_{b_k}$. The trailing blocks are $Z_u, Z_v \in O_{d-r}$. We study these ambiguities and their impact on the solution in the next section.*

Comment 4 (The unconstrained problem and joint-rank deficiency)**.** *We define the unconstrained problem related to a DCAT problem or formulation as the problem obtained by ignoring the determinant constraint, hence consisting in solving for a general affine transformation. The solution to the unconstrained problem related to formulation (15) is $A'_{\text{unc}} = \Gamma$, hence $A_{\text{unc}} = Q'' P''^{-1}$. We can thus make two observations; first, the unconstrained solution is unique, as the SVD ambiguities do not propagate to the estimated transformation A_{unc}; second, the rank of the estimated transformation A_{unc} coincides with the joint rank r. Hence, joint-rank deficiency causes degeneracy in the estimate. In contrast, in the DCAT problem, the rank of the estimated transformation A is always full. The next section shows that joint-rank deficiency nevertheless has an impact, obviously different from the unconstrained case, resulting in ambiguities in A.*

2.4 Polynomial System and Solution Derivation

We derive our solution from the orthogonal formulation (15).

2.4.1 General System and Strategy

The Lagrangian of the orthogonal formulation (15) is:

$$\mathcal{L} = \frac{1}{2} \left\|A' - \Gamma\right\|^2 + \lambda \left(\det(A') - s'\right), \tag{17}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. The first-order optimality conditions for A' and λ give:

$$\begin{cases} A' + \lambda A'^\star = \Gamma & \text{(18a)} \\ \det(A') = s', & \text{(18b)} \end{cases}$$

where $A'^\star = \det(A')(A')^{-\top}$ is a transposed adjugate. System (18) involves $d^2 + 1$ equations and unknowns. It is nonlinear and not trivially solvable. We transform it in a way which enables the derivation of the proposed two-step solution method. First, using the SVD theorem (theorem 1.2.1 in (Björck, 1996)), we introduce $A' = \tilde{U}_a \tilde{\Sigma} \tilde{V}_a^\top$, where $\tilde{U}_a, \tilde{V}_a \in O_d$ contain the left and right singular vectors respectively and $\tilde{\Sigma}$ is a diagonal matrix containing the non-negative singular values in non-increasing order. As A' is the main unknown we wish to resolve, this means that it will be replaced by the three SVD factors. The proposed solution method thus recovers these factors; however, it does not naturally satisfy the non-negativity and the non-increasing ordering requirements. Both requirements could be enforced as post-processing, to obtain an SVD *stricto sensu*, without changing the recomposition. This is however superfluous, given that the final matrix A' would be the same. We thus relax these requirements: we simply replace the SVD by its equivalent 'pseudo-SVD' $A' = U_a \Sigma V_a^\top$ where, for some diagonal sign matrices $S_u = \mathrm{diag}(\pm 1, \ldots, \pm 1) \in O_d$, $S_v = \mathrm{diag}(\pm 1, \ldots, \pm 1) \in O_d$, and a permutation matrix $G \in O_d$, $U_a = \tilde{U}_a G^\top S_u$, $V_a = \tilde{V}_a G^\top S_v$ and $\Sigma = S_u G \tilde{\Sigma} G S_v$. The pseudo-SVD is similar to the SVD but does not enforce the non negativity and ordering of the singular values. In other words, the pseudo-SVD is an SVD where the columns of U_a and V_a, and the diagonal elements of Σ, may be re-ordered and negated. Upon introducing the pseudo-SVD, we have $A'^\star = \det(\Sigma) U_a \Sigma^{-1} V_a^\top$ and $\det(A') = \det(\Sigma)$, leading to the system:

$$\begin{cases} U_a \left(\Sigma + \lambda \det(\Sigma)\Sigma^{-1}\right) V_a^\top = \Gamma & \text{(19a)} \\ \det(\Sigma) = s'. & \text{(19b)} \end{cases}$$

Second, we introduce the diagonal matrix $\Delta \in \mathbb{R}^{d \times d}$ and arrive at:

$$\begin{cases} U_a \Delta V_a^\top = \Gamma & \text{(20a)} \\ \Delta = \Sigma + \lambda \det(\Sigma)\Sigma^{-1} & \text{(20b)} \\ \det(\Sigma) = s'. & \text{(20c)} \end{cases}$$

We break down the system into two parts. Part 1) is equation (20a) and part 2) is a subsystem formed by equations (20b) and (20c). Concretely, our solution method first solves part 1) to retrieve U_a, V_a and Δ, and then solves part 2) to retrieve Σ given Δ. However, our derivation proceeds the other way around, meaning part 2) first, then part 1), as the analysis of part 1) depends on results obtained in the analysis of part 2). We provide the following lemma, which is used for solving system part 2).

Lemma 2 (Sign and order of the elements of Δ from system (20))**.** *We have* $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_d \geq 0$*, where* $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_d)$.

The proof of lemma 2 requires elements from the solution of part 1) of system (20). It is thus given after the system's solution method derivation, in section 2.4.3.

2.4.2 System Part 2)

We start by stating the polynomial formulation and studying its theoretical properties. We then derive specific solution methods for $d = 2$ and $d = 3$, for which we could obtain an explicit univariate polynomial, and for $d \geq 3$, for which we give a general numerical method based on polynomial resultants.

Polynomial formulation, properties and solution existence. System part 2) involves $d+1$ equations and unknowns. Upon multiplying equation (20b) by Σ and substituting equation (20c), it is equivalent to:

$$\begin{cases} \Sigma^2 - \Delta\Sigma + \lambda s' \mathrm{I} = 0 & \text{(21a)} \\ \det(\Sigma) = s'. & \text{(21b)} \end{cases}$$

This is a polynomial system whose d first equations are quadratic and whose last equation is of degree d. While this system has an elegant and compact form, its solution complexity quickly grows with d. Equation (21a) can be easily broken down into d *base equations* of the following form, for $i \in [1, d]$:

$$\sigma_i^2 - \delta_i \sigma_i + \lambda s' = 0. \tag{22}$$

We define the discriminant of the quadratic (22) as a function of λ as $D_i(\lambda) = \delta_i^2 - 4\lambda s'$. We express its two solutions as functions of λ as:

$$\sigma_i^\pm(\lambda) = \left(\delta_i \pm \sqrt{D_i(\lambda)}\right)/2. \tag{23}$$

A quadratic does not always have real solutions, and it is thus not trivial whether system (21) always has a real solution. For an isolated base equation, real solutions occur if and only if $D_i(\lambda) \geq 0$, hence if and only if $\delta_i^2 \geq 4\lambda s$. The solutions to system (21) given later in this section for several values of d confirm this point, as they always involve the roots of even-degree polynomials, which cannot be guaranteed to have real solutions. Proposition 1 below nonetheless shows that system (21) always possesses a real solution. We first provide a property revealing the sign-flipping of the solutions against the sign-flipping of δ_i and two lemmas on the solution signs used in the proof of proposition 1.

Property 1 (Solutions to the base equations upon sign-flips in Δ)**.** *The solution set to the base equation when flipping the sign of δ_i is obtained by flipping the sign of the original solutions. More precisely, we have, for $\delta_i \to -\delta_i$, that the solutions transform as $\sigma_i^{\pm}(\lambda) \to -\sigma_i^{\mp}(\lambda)$.*

Proof. We observe that $\delta_i \to -\delta_i$ does not change the discriminant $D_i(\lambda)$. We thus have $\sigma_i^{\pm}(\lambda) = \left(\delta_i \pm \sqrt{D_i(\lambda)}\right)/2 \to \left(-\delta_i \pm \sqrt{D_i(\lambda)}\right)/2 = -\left(\delta_i \mp \sqrt{D_i(\lambda)}\right)/2 = -\sigma_i^{\mp}(\lambda)$. □

Lemma 3 (Sign of the positive solution to the base equations)**.** *We have $\sigma_i^+(\lambda) > 0$.*

Proof. We lighten notation in the proof by not writing the dependency on λ explicitly. We first prove the non-negativity. We have $\sigma_i^+ = \delta_i + \sqrt{D_i}$. Because $\delta_i \geq 0$, from lemma 2, we have $\sigma_i^+ \geq 0$. We then prove the positivity. We have to show $\sigma_i^+ \neq 0$, which we do by contradiction. We have $\sigma_i^+ = 0 \Leftrightarrow \delta_i + \sqrt{D_i} = 0 \Leftrightarrow \delta_i = -\sqrt{D_i}$. As $\delta_i \geq 0$, the only possible solution is thus $\delta_i = 0 \wedge D_i = 0$. We expand the second term as $D_i = 0 \Leftrightarrow \delta_i^2 = 4\lambda s'$, which, from the first term, is $4\lambda s' = 0$, which, because $s' \neq 0$, is $\lambda = 0$. However, we have from equation (20b) that this implies $\Delta = \Sigma$, hence from equation (20c) that $\det(\Delta) = s' \Leftrightarrow \delta_1 \cdots \delta_d = s' \Leftrightarrow 0 = s'$, which contradicts $s' \neq 0$. □

Lemma 4 (Sign of the negative solution to the base equations)**.** *We have $\sigma_i^-(\lambda) = 0 \Leftrightarrow \lambda = 0$ and $\sigma_i^-(\lambda) < 0 \Leftrightarrow \operatorname{sign}(\lambda) \neq \operatorname{sign}(s')$.*

Proof. We lighten notation in the proof by not writing the dependency on λ explicitly. First, we have $\sigma_i^- = 0 \Leftrightarrow \delta_i - \sqrt{D_i} = 0 \Leftrightarrow \delta_i = \sqrt{D_i}$. Because $\delta_i \geq 0$, from lemma 2, and substituting the expression for D_i, this is equivalent to $\delta_i^2 = \delta_i^2 - 4\lambda s' \Leftrightarrow 4\lambda s' = 0 \Leftrightarrow \lambda = 0$, as $s' \neq 0$. Second, we have $\sigma_i^- < 0 \Leftrightarrow \delta_i - \sqrt{D_i} < 0 \Leftrightarrow \delta_i < \sqrt{D_i}$. Because $\delta_i \geq 0$, and substituting the expression for D_i, this is equivalent to $\delta_i^2 < \delta_i^2 - 4\lambda s' \Leftrightarrow 4\lambda s' < 0 \Leftrightarrow \operatorname{sign}(\lambda) \neq \operatorname{sign}(s')$. □

Proposition 1 (Existence of real solutions to system part 2))**.** *The polynomial system (21) has a real solution.*

Proof of proposition 1. For $d = 1$, the system has a trivial real solution $\sigma_1 = s'$ from equation (21b) and $\lambda = (\delta_1 - \sigma_1)\sigma_1/s' = \delta_1 - s'$ by substituting σ_1 in equation (22). As $s' \in \mathbb{R}_*$ we have $\sigma_1 \in \mathbb{R}_*$ and $\lambda \in \mathbb{R}$.

The reasoning for $d \geq 2$ is more involved. Our approach is to use the solution (23) for σ_i to the quadratic base equation (22) as a function of λ, formulate the conditions required for the existence of a real solution for σ_i, substitute the solution in the determinant equation (21b), which will then only depend on λ, and eventually show that a solution for λ satisfying the conditions required for the existence of a real solution to all the quadratics always exists.

We start with the case $s' > 0$. We have that there is a real solution to equation (22) if and only if $D_i(\lambda) \geq 0$, hence if and only if $\lambda \leq \delta_i^2/(4s')$. Because $\delta_1 \geq \cdots \geq \delta_d \geq 0$, from lemma 2, all the quadratics (22) for $i \in [1, d]$ have a real solution if and only if $\lambda \leq \delta_d^2/(4s')$. We define the product $\sigma_i \cdots \sigma_d$, which is the left-hand side of equation (21b), as a real function ϕ of λ, where each σ_i is given by its solution (23) from the corresponding quadratic, as $\phi(\lambda) = \prod_{i=1}^{d} \sigma_i^{\pm}(\lambda)$. Substituting in equation (21b), we need to show that $\phi(\lambda) = s'$ has a solution $\lambda \leq \delta_d^2/(4s')$. Function ϕ is defined on $(-\infty; \delta_d^2/(4s')]$ and is smooth on $(-\infty; \delta_d^2/(4s'))$. It depends on a sequence defining the sign of the solution to each quadratic. We choose this sequence so that it contains at least one negative sign and an even number of negative signs, which, because $d \geq 2$, is always possible. This particular sequence is a sufficient, but not necessary, condition for a real solution. We have $\lim_{\lambda \to -\infty} \phi(\lambda) = \infty$. This is because $\lim_{\lambda \to -\infty} D_i(\lambda) = \infty$, hence $\lim_{\lambda \to -\infty} \sigma_i^{\pm}(\lambda) = \pm\infty$ and the product of the sequence of signs is positive, from the even number of negative signs. We have $\phi(0) = 0$, as for the solution with a negative sign, $\sigma_i^-(0) = 0$ from lemma 1. As $s' > 0$, we have from the intermediate value theorem that $\phi(\lambda) = s'$ has a real solution on $(-\infty; 0) \subset (-\infty; \delta_d^2/(4s'))$.

We proceed with the case $s' < 0$. We have that there is a real solution to the base quadratic equation (22) if and only if $\lambda \geq \delta_i^2/(4s')$. Because $\delta_1 \geq \cdots \geq \delta_d \geq 0$, from lemma 2, all the quadratics (22) for $i \in [1, d]$ have a real solution if and only if $\lambda \geq \delta_d^2/(4s')$. Substituting in equation (21b), we need to show that $\phi(\lambda) = s'$ has a solution $\lambda \geq \delta_d^2/(4s')$. Function ϕ is defined on $[\delta_d^2/(4s'); \infty)$ and is smooth on $(\delta_d^2/(4s'); \infty)$. We choose the sequence defining the sign of the solution to each quadratic as containing at least one negative sign and an odd number of negative signs, which, because $d \geq 2$, is always possible. Hence, we have $\lim_{\lambda\to\infty} \phi(\lambda) = -\infty$ and $\phi(0) = 0$. As $s' < 0$, we have from the intermediate value theorem that $\phi(\lambda) = s'$ has a real solution on $(0; \infty) \subset (\delta_d^2/(4s'); \infty)$. □

Specific solution for $d = 2$. We specialise the general polynomial system (21) to the case $d = 2$, which has three unknowns, σ_1, σ_2 and λ, and the following three quadratic equations:

$$\begin{cases} q_1(\sigma_1, \lambda) = \sigma_1^2 - \delta_1\sigma_1 + \lambda s' = 0 \\ q_2(\sigma_2, \lambda) = \sigma_2^2 - \delta_2\sigma_2 + \lambda s' = 0 \\ q_0(\sigma_1, \sigma_2) = \sigma_1\sigma_2 - s' = 0. \end{cases} \tag{24}$$

We first eliminate σ_2 to form a simplified equation in σ_1 and λ, from which we solve for σ_1 as a function of λ. Equation q_0 gives $\sigma_2 = s'/\sigma_1$, which we substitute in equation q_2 and divide by λ, giving $\sigma_1^2 - \frac{\delta_2}{\lambda}\sigma_1 + \frac{s'}{\lambda} = 0$. Subtracting from equation q_1, we have $\sigma_1\left(\frac{\delta_2}{\lambda} - \delta_1\right) + \lambda s' - \frac{s'}{\lambda} = 0$, from which we arrive at the following quadratic-linear rational expression for σ_1 as:

$$\sigma_1 = s'\frac{\lambda^2 - 1}{\delta_2 - \lambda\delta_1}. \tag{25}$$

Following the same reasoning for σ_2, we obtain the dual expression:

$$\sigma_2 = s'\frac{\lambda^2 - 1}{\delta_1 - \lambda\delta_2}. \tag{26}$$

Substituting both expressions in equation q_0, we obtain $s'\frac{\lambda^2-1}{\delta_2-\lambda\delta_1}\frac{\lambda^2-1}{\delta_1-\lambda\delta_2} = 1$, and, upon expansion and simplification, the following depressed quartic in λ:

$$s'\lambda^4 - (2s' + \delta_1\delta_2)\lambda^2 + (\delta_1^2 + \delta_2^2)\lambda + s' - \delta_1\delta_2 = 0. \tag{27}$$

A Groebner basis approach gives the same result.

Specific solution for $d = 3$. We specialise the general polynomial system (21) to the case $d = 3$, which has four unknowns, σ_1, σ_2, σ_3 and λ, and the following four equations, which are three quadratics and one cubic:

$$\begin{cases} q_1(\sigma_1, \lambda) = \sigma_1^2 - \delta_1\sigma_1 + \lambda s' = 0 \\ q_2(\sigma_2, \lambda) = \sigma_2^2 - \delta_2\sigma_2 + \lambda s' = 0 \\ q_3(\sigma_3, \lambda) = \sigma_3^2 - \delta_3\sigma_3 + \lambda s' = 0 \\ q_0(\sigma_1, \sigma_2, \sigma_3) = \sigma_1\sigma_2\sigma_3 - s' = 0. \end{cases} \tag{28}$$

We used the Matlab Symbolic toolbox to compute the Groebner basis (Cox et al., 2005) of the ideal formed from the polynomials $\{q_1, q_2, q_3, q_0\}$ in the variables σ_1, σ_2, σ_3 and λ for the symbolic coefficients δ_1, δ_2, δ_3 and s' using lexicographic monomial ordering. The basis is composed of four polynomials:

$$\{f_1(\sigma_1, \lambda) = \sigma_1 + p_1(\lambda),\ f_2(\sigma_2, \lambda) = \sigma_2 + p_2(\lambda),\ f_3(\sigma_3, \lambda) = \sigma_3 + p_3(\lambda),\ f_4(\lambda)\}, \tag{29}$$

where p_1, p_2, p_3 and f_4 are univariate polynomials in λ. In particular, f_4 is a dodecic (a degree 12 polynomial) in λ, and thus the ideal has 12 zeros. Its symbolic coefficients are given in appendix A. Since f_1, f_2 and f_3 are linear in σ_1, σ_2 and σ_3, the original system of polynomials has a maximum of 12 solutions. The polynomials p_1, p_2 and p_3 are undecics (degree 11 polynomials). Their coefficients are rational functions of multivariate polynomials in δ_1, δ_2, δ_3 and s'; the numerators have degree lower or equal to 16 and the denominators have degree lower or equal to 13. The solutions are found numerically by finding the roots of f_4 in λ and substituting in f_1, f_2, f_3 to find the other unknowns.

General solution for $d > 3$. For $d > 3$, the Groebner basis with symbolic coefficients cannot be found with a symbolic solver.[1] We thus use a different strategy based on polynomial resultants. We proceed by finding a univariate polynomial in λ whose roots are guaranteed to contain the solutions. We first define the set of $d+1$ polynomial equations in the variables $\sigma_1, \ldots, \sigma_d, \lambda$, which are d quadratics and 1 degree-d polynomial, as:

$$\begin{cases} q_1(\sigma_1, \lambda) = \sigma_1^2 - \delta_1\sigma_1 + \lambda s' = 0 \\ \quad\vdots \\ q_d(\sigma_d, \lambda) = \sigma_d^2 - \delta_d\sigma_d + \lambda s' = 0 \\ q_0(\sigma_1, \ldots, \sigma_d, \lambda) = \sigma_1 \cdots \sigma_d - s' = 0. \end{cases} \tag{30}$$

By setting $f_0 = q_0$, we recursively define f_k as the following polynomial:

$$f_k(\sigma_k, \ldots, \sigma_d, \lambda) = R_{\sigma_k}[q_k, f_{k-1}], \quad k > 0, \tag{31}$$

where $R_x[p, q]$ denotes the polynomial resultant between p and q, and x is the eliminated variable. By following the recursive rule (31), f_d is a univariate polynomial in λ. For instance, for $d = 4$, f_d is of degree 32. Importantly, the resultant can be found with the polynomial coefficients $\delta_1, \ldots, \delta_d, s$ as symbolic variables.

The process of using resultants to eliminate variables in the system generates multiple spurious solutions. Each root of f_d requires checking its compatibility in the original system. This is usually a non-trivial task in a general system. However, since q_i for $i \in [1, d]$ is the base equation, it is quadratic in σ_i and gives the two solutions (23) independently for each σ_i, for each numerical solution for λ. We can thus compute 2^d sets of values for $\sigma_1, \ldots, \sigma_d$ for each real root of f_d, and then use q_0 for testing their compatibility with the original system.

2.4.3 System Part 1)

System part 1) is equation (20a). Given that matrices $U_a, V_a \in O_d$ and that matrix Δ is diagonal, by definition, this equation represents a pseudo-SVD of the ordered diagonal matrix Γ. Recall that the pseudo-SVD is an SVD ignoring the non-negativity and non–increasing ordering constraints on the singular values. The solution stems from the uniqueness of the singular values (Björck, 1996, §1.2.1)) and follows a similar pattern to that in comment 3 for the ambiguities of matrix Q''. For a set of simple singular values in Q'', which occurs when the diagonal values in Γ are all distinct and non-zero, the equation is solved by $\Delta = \Gamma$ and $U_a = V_a = S$, where $S = \operatorname{diag}(\pm 1, \ldots, \pm 1) \in O_d$. This is because the singular values of the diagonal matrix Γ are its diagonal elements. We ignore reordering, as it would be carried over to Σ and cancelled when recomposing matrix A'. Matrix S models the sign ambiguity occurring on the singular vectors (Björck, 1996, §1.2.1). For a set of singular values with multiple or vanishing singular values, the equation is solved by $\Delta = \Gamma$ and $U_a = \operatorname{diag}(Z, Z_x)$ and $V_a = \operatorname{diag}(Z, Z_y)$, where Z, Z_x, Z_y are the orthonormal ambiguity matrices occurring in the SVD of Q''. We ought to understand which of these ambiguities are carried over to the result in A', then in A, or cancel. Importantly, the cost is invariant to these ambiguities, hence cannot be used to resolve them.

We begin with the full-rank part, with ambiguities due to multiple singular values in Γ. For a block $Z_k \in O_{b_k}$, equation (23) shows that there can be two solutions to the σ_i entries, of the form $\sigma_i^{\pm}(\lambda) = \frac{1}{2}\left(\delta_i \pm \sqrt{D_i(\lambda)}\right)$. We consider without loss of generality that they are ordered and hence form two groups. The first group, of size c_k, contains the solution $\sigma_i^{+}(\lambda)$ and the second group, of size $b_k - c_k$, contains the solution $\sigma_i^{-}(\lambda)$, with $c_k \in [0, b_k]$. When forming $A' = U_a \Sigma V_a^{\top}$, the ambiguity on this block, denoted A'_k, is thus written as:

$$A'_k = \frac{1}{2}\hat{Z}_k\left(\delta_i \mathrm{I} + \sqrt{D_i(\lambda)}\,\tilde{\mathrm{I}}\right)\hat{Z}_k^{\top}, \tag{32}$$

where $\tilde{\mathrm{I}} = \operatorname{diag}(\mathrm{I}_{c_k}, -\mathrm{I}_{b_k - c_k})$. We need to count the number of Degrees of Freedom (DoF) d_k left in this block, which is shown to be equal in the worst case to the number of DoF in Z_k for $b_k = 2$ and strictly lower

[1] For $d = 4$ the basis was not found after 12 hours of computation on a high-end computer.

otherwise. Expanding, we have:

$$A'_k = \frac{1}{2}\left(\delta_i \mathrm{I} + \sqrt{D_i(\lambda)} Z_k \tilde{\mathrm{I}} Z_k^\top\right), \tag{33}$$

so it is about counting the number of DoF in the second term. This number is given by subtracting the number of DoF from the two orthonormal recombinations related to the two blocks in $\tilde{\mathrm{I}}$ to the number of DoF of Z_k, giving:

$$e_k = \frac{1}{2}\left(b_k(b_k-1) - c_k(c_k-1) - (b_k-c_k)(b_k-c_k-1)\right) = c_k(b_k-c_k).$$

For instance, with $c_k = 0$ or $c_k = b_k$, we have $e_k = 0$, as expected, and e_k grows as the combination of the two solutions reaches an equilibrium at $e_k = \lfloor b_k/2 \rfloor \lceil b_k/2 \rceil$ for $c_k = \lfloor b_k/2 \rfloor$. We thus have a varying number of DoF, depending on which solution is feasible and selected. The set of orthogonal matrices representing the ambiguity per multiple σ_i entry is non trivial to define. As it is a subset of O_{b_k}, and as any matrix from O_{b_k} leaves the cost invariant, we simply use O_{b_k} as the ambiguity set.

We proceed with the rank-deficient part, with ambiguities due to vanishing values in Γ but not vanishing in Σ. The solution for the concerned σ_i entries are $\sigma_i^\pm(\lambda) = \pm\sqrt{\lambda s'}$, which also shows that $r < d$ implies $\lambda \neq 0$, meaning that the determinant constraint is always active. The recombined block in A' is thus $A'_k = Z_u \bar{\Sigma} Z_v^\top = \sqrt{\lambda s'} Z_u \operatorname{diag}(\pm 1, \ldots, \pm 1) Z_v^\top$, which is a scaled orthonormal matrix $\sqrt{\lambda s'}\hat{Z}$, with $\hat{Z} \in O_{d-r}$ endowed with the corresponding number of DoF $(d-r)(d-r-1)/2$.

In summary, we arrive at a simplified ambiguity structure expressed by $U_a = V_a = \hat{K}$ in the general case, with:

$$\hat{K} = \operatorname{diag}(Z_1, \ldots, Z_l, \hat{Z}),\ Z_k \in O_{b_k} \text{ for } k \in [1, l] \text{ and } \hat{Z} \in O_{d-r}. \tag{34}$$

We end this section with the following proof, pending from section 2.4.1.

Proof of lemma 2. The solution to part 1) of system (20) given directly above in section 2.4.3 leads to $\Delta = \Gamma$. As Γ is obtained from an SVD, it contains singular values, which are non-negative and ordered. The same properties thus directly apply to $\delta_1, \ldots, \delta_d$ forming the diagonal elements of Δ. □

2.5 Solution Procedure

We give procedure 1 which implements the proposed method, following the derivation steps presented above. This procedure can be used as pseudo-code to implement DCAT in a programming language. Step 1) standardises the problem following section 2.1, step 2) reduces it to virtual point pairs following section 2.2, step 3) orthogonalises it to a diagonal form following section 2.3, from which the polynomial system (20) is readily available. The RQ factorisation in step 2) is implemented by the cheaper thin-RQ factorisation via Cholesky decomposition (Terao et al., 2020), which does not compute the unused B_2 factor. In can be obtained in closed-form for $d \in \{2, 3\}$. The SVD in step 3) can be obtained in closed-form for $d = 2$. Step 4) solves the polynomial system following section 2.4 and steps 5) and 6) 'undo' the initial simplifications to retrieve the solution. Eventually, the optional step 7) identifies the ambiguities using the optional input parameter τ to decide the numerically repeated singular values and the numerical joint rank.

3 Prescribed Volume and Orientation Simplex

We start with specifying the PVOS problem statement, from which we show an equivalence with the minimal DCAT problem. We then derive a specific solution. Finally, we give a computational procedure which implements the proposed method in pseudo-code.

3.1 Problem Statement

The PVOS problem is stated in equation (3), which involves the signed volume function $\mathcal{W}$ of simplex N, given by:

$$\mathcal{W}(N) = \frac{1}{d!} \det([N^\top\, e]). \tag{35}$$

Algorithm 1 Computational procedure solving DCAT. Parameter τ is only needed to generate the complete solution set if ambiguities arise. These ambiguities are decided from the numerical closeness between singular values and the numerical τ-rank defined in (Björck, 1996, §2.7.1). The computational structure chosen to hold the ambiguities is left to the user's discretion.

```
⇒ Function DCAT($P, Q \in \mathbb{R}^{d\times m},\ w \in \mathbb{R}_*^{m\times 1},\ s \in \mathbb{R}_*$, (optional) $\tau \in \mathbb{R}_+$)
1) Standardise the problem by removing translation and weighting:
    • Set the diagonal weight matrix as $W = \operatorname{diag}(w)$
    • Compute the weighted centroids of each point set as $\bar{q} = \frac{1}{\|W\|^2} QW^2 e$ and $\bar{p} = \frac{1}{\|W\|^2} PW^2 e$
    • Compute the centred weighted point sets as $P' = \left(P - \bar{p}e^\top\right) W$ and $Q' = \left(Q - \bar{q}e^\top\right) W$
2) Reduce the problem to $d$ virtual point pairs:
    • Compute the thin RQ factorisation $P' = RB_1^\top$
    • Set $P'' = R$ and $Q'' = Q'B_1$
3) Orthogonalise the problem to produce a diagonal form:
    • Compute the SVD $Q'' = U\Gamma V^\top$
    • Set $s' = s\det(R)\det(U)\det(V)$
4) Solve the polynomial system:
    • Extract $\gamma_1, \dots, \gamma_d$ from the diagonal matrix $\Gamma$
    • For $d = 2$, solve the quartic (27) for $\lambda$ and select the real solutions with minimal cost $(\sigma_1 - \gamma_1)^2 + (\sigma_2 - \gamma_2)^2$ by obtaining $\sigma_1$ and $\sigma_2$ from equations (25,26) (per proposition 1, there are between 1 and 4 such real solutions)
    • For $d = 3$, solve the dodecic $f_4(\lambda) = 0$ in equation (29) for $\lambda$ and select the real solutions with minimal cost $(\sigma_1 - \gamma_1)^2 + (\sigma_2 - \gamma_2)^2 + (\sigma_3 - \gamma_3)^2$ by obtaining $\sigma_1$, $\sigma_2$ and $\sigma_3$ from the polynomials $f_1$, $f_2$ and $f_3$ in equation (29) (per proposition 1, there are between 1 and 12 such real solutions)
    • For $d > 3$, solve the univariate polynomial $f_d(\lambda) = 0$ for $\lambda$, form the $2^d$ solution sets for $\sigma_1, \dots, \sigma_d$ from equation (23) for each value of $\lambda$ and keep the solutions which satisfy $q_0(\sigma_1, \dots, \sigma_d, \lambda) = 0$ and minimise the cost $\sum_{i=1}^d (\sigma_i - \gamma_i)^2$ (per proposition 1, there is at least 1 such real solution)
    • Assemble each solution $\sigma_1, \dots, \sigma_d$ to a diagonal matrix $\Sigma$
5) Select one solution and retrieve the linear part as $A = U\Sigma V^\top R^{-1}$
6) Compute the translation as $t = \frac{1}{\|W\|^2}(Q - AP)W^2 e = \bar{q} - A\bar{p}$
7) (optional) Identify the possible ambiguities:
    • Compute the numerical joint $\tau$-rank $r$ as the number of singular values in $\Gamma$ greater than $\tau$
    • Compute the numerical repeated values in $\Gamma$, where a value is repeated if $|\gamma_i - \gamma_j| \leq \tau$, obtaining $b_1, \dots, b_l$ with $l \leq r$ and $\sum_{k=1}^l b_k = r$
    • For each solution, store the simplified ambiguity structure $\hat{K}$ following equation (34)
⇒ Output $A \in \mathbb{R}^{d\times d},\ t \in \mathbb{R}^{d\times 1}$, (optional) the entities required to generate all solutions
```

This definition allows us to derive the following relationship between the signed simplex volume and an affine transformation (A, t) of its points:

$$\mathcal{W}(AN + te^\top) = \frac{1}{d!}\det\left(\begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} N \\ e^\top \end{bmatrix}\right) = \frac{1}{d!}\det\left(\begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}\right)\det\left(\begin{bmatrix} N \\ e^\top \end{bmatrix}\right) = \det(A)\mathcal{W}(N). \tag{36}$$

This relationship shows that the signed volume changes according to the determinant of the affine transformation, which we exploit directly below to draw an equivalence between the PVOS and a particular instance of the DCAT problems.

3.2 Equivalence with Minimal Determinant-Constrained Affine Transformation

We first show how a PVOS problem can be transformed to a minimal DCAT problem and then vice-versa, showing the equivalence between the two problems.

3.2.1 Transformation of PVOS to Minimal DCAT

We start from the PVOS formulation (3) and perform a change of variable $N = AM + te^\top$, with (A, t) an unknown affine transformation:

$$\min_{(A,t)\in\mathbb{R}^{d\times d+1}} \|(AM + te^\top - M)W\|^2 \quad \text{s.t.} \quad \mathcal{W}(AM + te^\top) = h. \tag{37}$$

Using equation (36), we rewrite the constraint as:

$$\mathcal{W}(AM + te^\top) = \det(A)\mathcal{W}(M) = h, \tag{38}$$

hence as:

$$\det(A) = h/\mathcal{W}(M). \tag{39}$$

We arrive at the following DCAT problem:

$$\min_{(A,t)\in\mathbb{R}^{d\times d+1}} \|(AP + te^\top - Q)W\|^2 \quad \text{s.t.} \quad \det(A) = s,$$

with $P = Q = M$ and $s = h/\mathcal{W}(M)$. Once (A, t) is solved, N is directly obtained from the above change of variable. Importantly, this is a special DCAT problem because $P = Q$, for which there is a specific solution method.

3.2.2 Transformation of Minimal DCAT to PVOS

We start from a DCAT problem (1) in the minimal case $m = d + 1$ and perform a change of variable $N = AP + te^\top$ with $N \in \mathbb{R}^{d\times d+1}$ a d-simplex. Using equation (36), we have $\mathcal{W}(N) = \det(A)\mathcal{W}(P) = s\mathcal{W}(P)$. Upon substituting, we have:

$$\min_{N\in\mathbb{R}^{d\times d+1}} \|(N - Q)W\|^2 \quad \text{s.t.} \quad \mathcal{W}(N) = s\mathcal{W}(P). \tag{40}$$

We arrive at the following PVOS problem:

$$\min_{N\in\mathbb{R}^{d\times d+1}} \|(N - M)W\|^2 \quad \text{s.t.} \quad \mathcal{W}(N) = h, \tag{41}$$

with $M = Q$ and $h = s\mathcal{W}(P)$. Once N is solved, (A, t) is found by inverting the change of variable formula $N = [A\,t][P^\top\, e]^\top$, leading to $[A\,t] = N[P^\top\, e]^{-\top}$.

3.3 Specific Solution Derivation

PVOS, following the above-derived equivalence, is solved by the general DCAT method implemented by procedure 1. The specifics of the special minimal DCAT problem equivalent to the PVOS problem however allow one to derive a simplified and computationally more efficient specific solution method. We proceed by specialising each step of the general derivation of section 2.

We start from the PVOS problem and transform it to the equivalent minimal DCAT problem with $P = Q$ following section 3.2.1. We proceed with the standardisation as in section 2.1. We arrive at the specialised translation formula:

$$t = (\mathrm{I} - A)\bar{p}, \tag{42}$$

and the specialised standard problem formulation:

$$\min_A \left\|(A - \mathrm{I})P'\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{43}$$

We then perform the reduction and orthogonalisation in a single step with an SVD $P' = U\Theta V^\top$, with $U \in O_d$, $V \in O_{d+1}$, $\Theta = [\bar{\Theta}\, 0] \in \mathbb{R}^{d\times d+1}$ and $\bar{\Theta} \in \mathbb{R}^{d\times d}$ is a full-rank diagonal matrix. We introduce this decomposition in the specialised standard formulation (43), cancelling factor V thanks to the L_2 norm invariance to unitary-matrix multiplication, obtaining the specialised reduced formulation:

$$\min_A \left\|(A - \mathrm{I})U\bar{\Theta}\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{44}$$

Expanding and pre-multiplying by $U^\top$, we obtain:

$$\min_A \left\|U^\top A U\bar{\Theta} - \bar{\Theta}\right\|^2 \quad \text{s.t.} \quad \det(A) = s. \tag{45}$$

Upon changing variable by introducing $A' = U^\top A U\bar{\Theta}$, we arrive at the specialised orthogonal formulation:

$$\min_{A'} \left\|A' - \bar{\Theta}\right\|^2 \quad \text{s.t.} \quad \det(A') = s', \tag{46}$$

where $s' = \det(A') = \det(U^\top A U\bar{\Theta}) = s\det(\bar{\Theta})$. The specialised orthogonal formulation (46) is equivalent to the orthogonal formulation (15) of a general DCAT problem and may thus be solved similarly, following the polynomial formulation and solution methods of section 2.4. A minor difference lies in the ambiguities, which are restricted as $\bar{\Theta}$ does not contain vanishing singular values. Once A' is solved, we obtain the sought solution for A by inverting the change of variable as:

$$A = UA'\bar{\Theta}^{-1}U^\top. \tag{47}$$

3.4 Solution Procedure

We give procedure 2 which implements the proposed method, following the derivation steps presented above. This procedure can be used as pseudo-code to implement PVOS in a programming language. The PVOS problem is implicitly cast into a special minimal DCAT problem. Step 1) standardises the problem following section 2.1, step 2) reduces it to virtual simplex points and orthogonalises it to a diagonal form following section 3.3, from which the polynomial system (20) is readily available. The SVD of step 2) is implemented by a cheaper thin SVD with an eigen-decomposition of $M'M'^\top$, since factor V is not required (Demmel, 2020). This decomposition can be obtained in closed-form for $d = 2$. Step 3) solves the polynomial system following section 2.4 and steps 4), 5) and 6) 'undo' the initial simplifications to retrieve the solution. Eventually, the optional step 7) identifies the ambiguities using the optional input parameter τ to decide the numerically repeated singular values.

Algorithm 2 Computational procedure solving PVOS. Parameter τ is only needed to generate the complete solution set if ambiguities arise. These ambiguities are decided from the numerical closeness between singular values. The computational structure chosen to hold the ambiguities is left to the user's discretion.

```
⇒ Function PVOS($M \in \mathbb{R}^{d\times d+1},\ w \in \mathbb{R}_*^{d+1},\ h \in \mathbb{R}_*$, (optional) $\tau \in \mathbb{R}_+$)
1) Standardise the problem by removing translation and weighting:
    • Set the diagonal weight matrix as $W = \mathrm{diag}(w)$
    • Compute the weighted centroid of the point set as $\bar{m} = \frac{1}{\|W\|^2} M W^2 e$
    • Compute the centred weighted point set as $M' = \left(M - \bar{m}e^\top\right) W$
2) Reduce to $d$ virtual point pairs and orthogonalise the problem to a diagonal form:
    • Compute the thin SVD $M' = U\bar{\Theta}\bar{V}^\top$
    • Set $s' = h \det(\bar{\Theta}) / \mathcal{W}(M)$
3) Solve the polynomial system by following step 4) of the DCAT procedure 1 with $\Gamma = \bar{\Theta}$
4) Select one solution and retrieve the linear part as $A = UA'\bar{\Theta}^{-1}U^\top$
5) Compute the translation as $t = (I - A)\bar{m}$
6) Compute the simplex as $N = AP + te^\top$
7) (optional) Identify the possible ambiguities by following step 7) of the DCAT procedure 1 with $\Gamma = \bar{\Theta}$
   and $r = d$
⇒ Output $N \in \mathbb{R}^{d\times d+1}$, (optional) the entities required to generate all solutions
```

4 Application: Laparoscopic Augmented Reality

Laparoscopic surgery is a form of modern mini-invasive surgery, with navigation challenges for which guidance by augmented reality has recently brought promising results, particularly in liver surgery. Laparoscopic surgery and the registration problem related to augmented reality are advanced research fields. We give a short account of them and outline a registration method from the literature (Koo et al., 2017), which serves as a base for the proposed method exploiting DCAT and PVOS, and as baseline for experimental comparison.

4.1 Laparoscopic Surgery, Augmented Reality and the Registration Problem

In laparoscopic surgery, the surgeon uses a surgical camera introduced in the abdominal cavity through a tiny incision. The camera films the target organ, which may be the liver, the pancreas, the prostate, or the uterus, and the surgeon performs the surgical procedure by looking at the camera's image on a screen. This is technically challenging because the surgeon cannot directly touch the organ and cannot see within its volume where the tumours are, before incision. Augmented reality mitigates this challenge by overlaying a preoperative 3D model of the organ, including the tumours, on the surgical image, enhancing the surgeon's spatial understanding. The preoperative 3D model is a volumetric mesh reconstructed from a CT scan or an MRI acquired prior to surgery. CT and MRI can generally not be performed during surgery. The core difficulty to create an augmented reality system of this kind is to register the preoperative 3D model with the surgical 2D image. The registration moves, rotates, and deforms the preoperative 3D model so that it properly projects on the observed organ in the surgical image. We focus on the case of liver surgery, which is particularly difficult because of its substantial deformability. Nonetheless, the liver has distinct anatomical landmarks, which are exploited to constrain registration. The anatomical landmarks are corresponding curves visible both in the preoperative 3D model and in the surgical image: the falciform ligament and specific parts of the lower ridge.

4.2 Modelling and Base Registration Method

We use the state-of-the-art model and method for liver registration (Adagolodjo et al., 2017; Koo et al., 2017). We model the surgical camera by the pin-hole model (Hartley and Zisserman, 2003) and assume the intrinsic parameters, which are the focal length f and the principal point x_0, y_0 in pixels, to be known from static calibration. The camera's coordinate frame is centred at the camera's projection centre and oriented along the principal and retinal axes. A 3D point with coordinates (X, Y, Z) expressed in this coordinate frame is projected to the image as $(x, y) = \Pi(X, Y, Z) = (fX/Z + x_0, fY/Z + y_0)$. The preoperative 3D model is expressed in its own coordinate frame, defined by the CT or MRI acquisition device. The objective of registration is to compute the registration function $\phi : \mathbb{R}^3 \to \mathbb{R}^3$, which maps points from the preoperative coordinate frame to the camera coordinate frame. The preoperative 3D model is a 3D mesh represented by a set of n vertices $\mathcal{V}_{\text{pre}} \in \mathbb{R}^{3\times n}$, a set of vertex triplets $\mathcal{D}_{\text{liv}} \subset [1, n]^3$ defining triangles at the liver surface and a set of vertex quadruplets $\mathcal{T}_{\text{liv}} \subset [1, n]^4$ defining tetrahedra spanning the liver volume. We assume without loss of generality that the liver contains a single tumour, whose surface is represented by a set of vertex triplets $\mathcal{D}_{\text{tum}} \subset [1, n]^3$. The ℓ anatomical landmarks are each represented by a set of indices $\mathcal{L}_k \subset [1, n]$, $k \in [1, \ell]$, defining piecewise linear curves at the liver surface. We denote as m the total number of landmark vertices. The corresponding observed anatomical landmarks in the surgical image are each represented by a list of pixel coordinates $\mathcal{C}_k \subset \mathbb{N}^2$, $k \in [1, \ell]$, defining continuous curves. The registration transformation ϕ is modelled by the displacement of the mesh vertices, from their location $\mathcal{V}_{\text{pre}}$ in the preoperative coordinate frame to their desired location $\mathcal{V}_{\text{sur}} \in \mathbb{R}^{3\times n}$ in the surgical camera coordinate frame. The transformation ϕ is constrained by the anatomical landmarks and prior assumptions on the admissible deformations. The assumptions used by existing methods range from rigidity, where ϕ is modelled by a rigid-body motion, to complex biomechanics. These models were implemented using the principle of position-based dynamics (Bender et al., 2015; Koo et al., 2017). We propose two volume-preserving models of ϕ based on DCAT and PVOS, which we implement in the same framework.

The principle is to consider the mesh vertices as particles, to which the registration constraints are cyclically applied until convergence. An initial step simply initialises the registration parameters $\mathcal{V}_{\text{sur}}$ with a rigid transformation of $\mathcal{V}_{\text{pre}}$ accounting for the global changes owing to the camera viewpoint. This is usually done by requesting the user to manually translate and rotate the preoperative 3D model in order to visually overlay its reprojection with the observed liver in the surgical image (Koo et al., 2017; Labrunie et al., 2022). We have automated this step by determining a rigid-body motion that brings the preoperative 3D model to a typical surgical camera pose, with the anterior part of the liver filling the surgical image and the liver ridge facing downward. We also also implemented the projection using rasterisation to handle visibility-aware mesh projection, rather than mere vertex-wise projection in the original implementation (Koo et al., 2017).

Concretely, the generic iterative algorithm then has three main steps repeated until convergence, which we describe next, following algorithm 3. Step 1) starts enforcing the registration constraints. It uses the anatomical landmarks by finding the 3D point pairs in matrices $P, Q \in \mathbb{R}^{3\times m}$ representing the correction that each landmark point should be subjected to. For each landmark of index k, and for each mesh vertex in $\mathcal{V}_{\text{sur}}$ forming this landmark, *i.e.*, with index in $\mathcal{L}_k$, it finds the closest 3D point projected by function Π on the observed landmark $\mathcal{C}_k$ in the surgical image. We store in P the current landmark vertex coordinates extracted from $\mathcal{V}_{\text{sur}}$ and in Q the closest 3D points. We also use the liver silhouette, whose vertices are included in the point pairs. Step 2) enforces the deformation prior, by computing new mesh vertices $\mathcal{V}'_{\text{sur}}$ from the current estimate $\mathcal{V}_{\text{sur}}$ and the 3D point pairs in P, Q. It depends on the chosen strategy for the deformation prior, user-specified in variable strategy. The first existing strategy is Rigid, which is a rigidity prior on transformation ϕ. One estimates a rigid-body motion between the points in P and the points in Q in the least-squares sense and applies this transformation to the vertices in $\mathcal{V}_{\text{sur}}$ to find $\mathcal{V}'_{\text{sur}}$. The second existing strategy, named Biomechanical, is the biomechanical formulation from (Koo et al., 2017). The two proposed strategies implement volume preserving transformations; they are described in the next section. Step 3) updates the vertices in $\mathcal{V}_{\text{sur}}$ using a momentum parameter $0 < \alpha \leq 1$ as $\mathcal{V}_{\text{sur}} \leftarrow (1-\alpha)\mathcal{V}_{\text{sur}} + \alpha\mathcal{V}'_{\text{sur}}$. Steps 1) to 3) are repeated until $\|\mathcal{V}_{\text{sur}} - \mathcal{V}'_{\text{sur}}\|/n < \epsilon$ or until the iteration count reaches 100. We use the recommended values of $\alpha = 0.2$ and $\epsilon = 10^{-2}$ in all our experiments.

4.3 Registration with Volume Preservation

We propose to use volume preservation on the transformation ϕ. We define two types of priors on ϕ and integrate them as strategies to the above-described generic registration method in algorithm 3.

Global volume preservation. We propose the GlobalVolumePreservation strategy, which is an affine transformation with global volume preservation prior. This is achieved by representing ϕ by a DCAT in dimension $d = 3$ with determinant $s = 1$ and unit weights. This prior is ‘softer’ than the rigidity prior, which also preserves the global volume, yet at the expense of being overly stiff, and is stiffer than the biomechanical models, which do not preserve the global volume, but may be underconstrained by the anatomical landmarks. Concretely, we estimate a DCAT (A, t) from the 3D point pairs in P, Q using the computational procedure in algorithm 1 and set $\mathcal{V}'_{\text{sur}} = A\mathcal{V}_{\text{sur}} + te^\top$. As DCAT forms a group, namely $SL_3(\mathbb{R})$, the algorithm guarantees that the estimated transformation ϕ is eventually a DCAT with determinant 1.

Local volume preservation. We propose the LocalVolumePreservation strategy, which is a local volume preservation prior. This is achieved by imparting a change of the mesh tetrahedra by a PVOS in dimension $d = 3$ with volume h taken from the original mesh $\mathcal{V}_{\text{pre}}$ and unit weights. Indeed, the elements of the volumetric 3D mesh are tetrahedra, which are 3-simplices. For each tetrahedron, we store the $d + 1 = 4$ vertices in $M \in \mathbb{R}^{3\times 4}$. We then compute the closest simplex $N \in \mathbb{R}^{3\times 4}$ with volume h measured from the corresponding tetrahedron in the preoperative 3D model with vertices $\mathcal{V}_{\text{pre}}$. This prior is softer than the global volume preservation prior, yet strongly constrains the deformations. Concretely, we use PVOS-corrected vertex coordinates N from M using the computational procedure in algorithm 2 and update the vertices in $\mathcal{V}'_{\text{sur}}$.

Algorithm 3 Computational procedure solving registration for laparoscopic augmented reality. The base algorithm is adapted from (Koo et al., 2017). The core step 2) estimates the registration transformation ϕ implicitly by updating the registered model vertices in $\mathcal{V}_{\text{sur}}$, depending on the chosen strategy, specified in the input variable strategy. The numerical hyperparameters α and ϵ represent momentum and convergence threshold.

```
⇒ Function registration($\mathcal{V}_{\text{pre}}, \mathcal{V}_{\text{sur}} \in \mathbb{R}^{3\times n},\ \mathcal{T}_{\text{liv}} \subset [1,n]^4,\ \mathcal{L}_1,\ldots,\mathcal{L}_\ell \subset [1,n],\ \mathcal{C}_1,\ldots,\mathcal{C}_\ell \subset \mathbb{N}^2$, strategy,
   (optional) $0 < \alpha \leq 1$ (default: $\alpha = 0.2$), $\epsilon \in \mathbb{R}_+$ (default: $\epsilon = 10^{-2}$))

   Set iteration count as iter = 0

1) Reset $P = \emptyset$ and $Q = \emptyset$, and set iter = iter + 1

   For each landmark of index $k = 1,\ldots,\ell$

     For each mesh vertex of index $j \in \mathcal{L}_k$

       Extract the point coordinates as $p = \mathcal{V}_{\text{sur}}(j)$

       Find the closest point $q$ projecting on the image landmark as $\min_{q\in\mathbb{R}^3} \|p - q\|^2$ s.t. $\Pi(q) \in \mathcal{C}_k$

       Store $p, q$ as $P = [P\,p]$ and $Q = [Q\,q]$

2) Use the point pairs in $P, Q$ to update the vertices in $\mathcal{V}_{\text{sur}}$, depending on strategy:

     • "Rigid":
         Compute the least-squares rigid transformation $(R, t)$ between $P$ and $Q$ (Umeyama, 1991)
         Set $\mathcal{V}'_{\text{sur}} = R\mathcal{V}_{\text{sur}} + te^\top$
     • "Biomechanical":
         Refer to (Koo et al., 2017) for details
     • "GlobalVolumePreservation":
         Compute $(A, t) = \text{DCAT}(P, Q, 1)$ from algorithm 1
         Set $\mathcal{V}'_{\text{sur}} = A\mathcal{V}_{\text{sur}} + te^\top$
     • "LocalVolumePreservation":
         For each tetrahedron with vertex indices $u_1,\ldots,u_4 \in [1,n]$ in $\mathcal{T}_{\text{liv}}$
           Extract original point coordinates from $\mathcal{V}_{\text{sur}}$ or the projection-corrected ones from $Q$ as $M$
           Extract the original tetrahedron volume from $\mathcal{V}_{\text{pre}}$ as $h$
           Compute the nearest volume-preserving tetrahedron as $N = \text{PVOS}(M, e, h)$ from algorithm 2
           Store the point coordinates from $N$ in $\mathcal{V}'_{\text{sur}}$

3) Update the registered model estimate as $\mathcal{V}_{\text{sur}} = (1-\alpha)\mathcal{V}_{\text{sur}} + \alpha\mathcal{V}'_{\text{sur}}$

4) Stopping criterion: if $\|\mathcal{V}_{\text{sur}} - \mathcal{V}'_{\text{sur}}\| \geq n\epsilon$ and iter < 100 goto step 1)

⇒ Output $\mathcal{V}_{\text{sur}} \in \mathbb{R}^{3\times n}$
```

4.4 Experimental Results

We report experimental results on a public dataset comprising 4 patient data (Rabbani et al., 2022). This dataset includes the ground truth localisation of the tumour in the camera coordinate frame, allowing one to measure the tumour registration error, providing sound means of comparing registration methods. This is achieved by using the estimated transformation ϕ to transport the tumour vertices given by $\mathcal{D}_{\text{tum}}$ from the preoperative to the surgical coordinate frames and comparing them to the ground truth tumour using least squares. We compare 7 methods. The first 5 are existing methods from the literature, forming the state-of-the-art. One of them, named `Rigid`, uses the principle described in section 4.2 with a rigid-body motion estimated in step 2). The last 2 methods, named `GloVP` and `LocVP`, are the global and local volume preservation methods given in section 4.3.

Method	**Patient 1**	**Patient 2**	**Patient 3**	**Patient 4**	**Avg**	**Avg w/o P2**
Manual Initialisation	15.14	35.48	30.48	16.29	24.35	20.64
(Adagolodjo et al., 2017)	8.25	37.25	28.40	15.83	22.43	17.49
(Koo et al., 2017)	9.49	38.95	25.04	18.35	22.95	17.63
(Labrunie et al., 2022)	14.84	N/A	22.40	7.23	N/A	14.82
`Rigid`	14.93	84.89	22.34	17.51	34.92	18.26
`GloVP`	17.38	69.07	12.17	14.74	28.34	14.76
`LocVP`	17.39	69.24	11.53	14.62	28.19	14.51

Table 1: Laparoscopic augmented reality: tumour registration error (mm) for the 7 compared methods.

We show quantitative results in table 1 for the tumour registration error and qualitative results in figure 2. The average error is shown for all 4 patients and excluding patient 2, which is known to incur higher errors in all methods, owing to a significant torsion of the liver. We observe that all methods indeed have a high error for patient 2 and use the average error without patient 2 as reference. The rigid methods, namely Manual Initialisation and `Rigid`, have the highest average errors, at 20.64mm and 18.26mm respectively. Next, methods (Adagolodjo et al., 2017; Koo et al., 2017) based on a deformable model have slightly lower average errors, at 17.49mm and 17.63mm respectively. Method (Labrunie et al., 2022) with visibility reasoning defines the state-of-the-art, with an average error at 14.82mm. Lastly, the proposed volume-preserving methods `GloVP` and `LocVP` have the lowest average errors, at 14.76mm and 14.51mm respectively.

5 Conclusion

We have shown that estimating a transformation in the $SL_d(\mathbb{R})$ or the $SL_d^{\pm}(\mathbb{R})$ groups from point pairs in the least-squares sense is achievable optimally in spite of the nonconvex determinant constraint. We have given a computation procedure reducing the problem to a univariate polynomial. We have carefully studied the conditions for a single or multiple solutions, establishing a simple relationship to the cases where estimating an affine transformation by forgetting the determinant constraint would yield a singular result. We have shown that the proposed study enables computing the closest volume-preserved simplex to a given d-simplex optimally, for which we proposed a specific computation procedure. This work simplifies and generalises previous work addressing the specific case of dimension $d = 2$.

Future work may follow numerous leads. These leads include *i)* the case of a determinant prescribed in a subset of $\mathbb{R}$, rather than as a single fixed value in $\mathbb{R}$, *ii)* the extension to projective transformations with prescribed determinant, either for the complete homography matrix or its leading sub-matrix, *iii)* the study of the manifold of d-simplices with prescribed volume, to which the proposed PVOS forms a projector, and *iv)* the use of other costs such as the symmetric least-squares cost occurring in maximum-likelihood estimators.

Acknowledgements. This work has been supported by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 through project METAMORPH (PID2023-151295OB-I00).

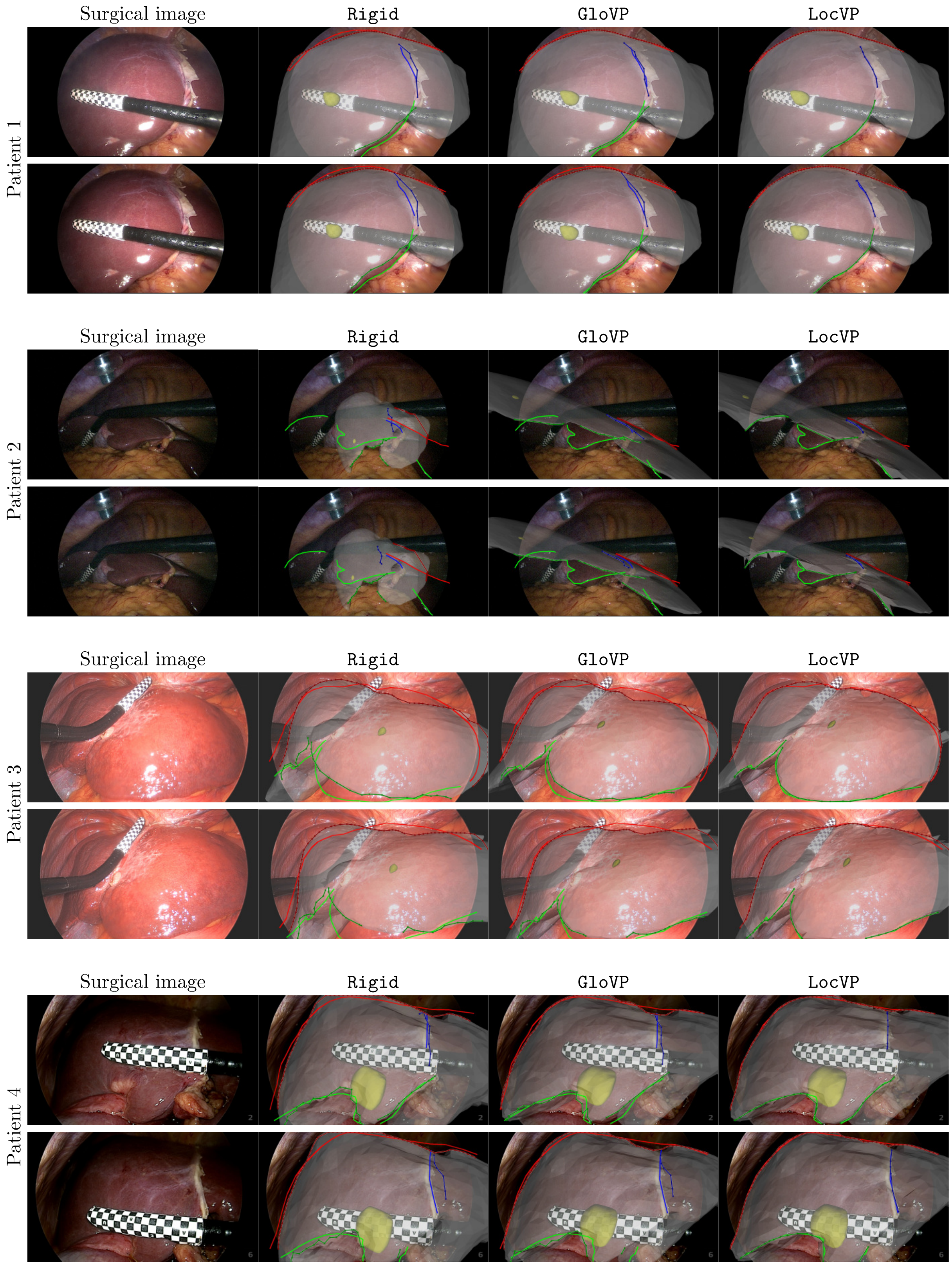


Figure 2: Laparoscopic augmented reality: registration visualisation. The existing method `Rigid` and the two proposed ones, `GloVP` based on DCAT, and `LocVP` based on PVOS are shown for two images per patient.

A Univariate Polynomial for $d = 3$

The proposed method to solve the polynomial system occurring in DCAT for dimension $d = 3$ may be reduced to the following univariate in the Lagrange multiplier λ:

$$f_4(\lambda) = a_{12}\lambda^{12} + \cdots + a_1\lambda + a_0, \tag{48}$$

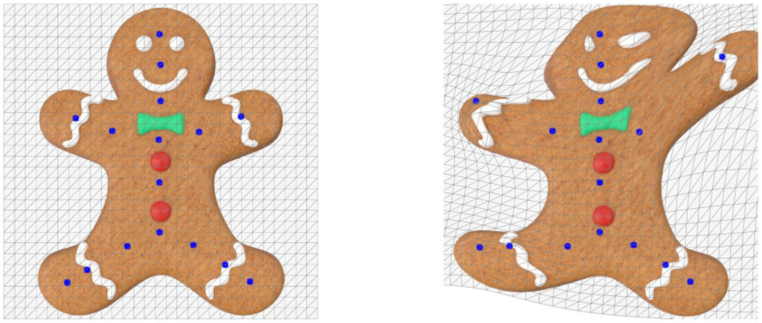
where:

$$\begin{aligned}
a_0 &= 1\\
a_1 &= 0\\
a_2 &= 0\\
a_3 &= \frac{4s' - \delta_1\delta_2\delta_3}{s'^2},\\
a_4 &= -\frac{2(\delta_1^2 + \delta_2^2 + \delta_3^2)}{s'^2}\\
a_5 &= \frac{\delta_1^2\delta_2^2 + \delta_1^2\delta_3^2 + \delta_2^2\delta_3^2}{s'^3},\\
a_6 &= \frac{6s' + 5\delta_1\delta_2\delta_3}{s'^3}\\
a_7 &= -\frac{\delta_1^3\delta_2\delta_3 + 4s'\delta_1^2 + \delta_1\delta_2^3\delta_3 + \delta_1\delta_2\delta_3^3 + 4s'\delta_2^2 + 4s'\delta_3^2}{s'^4}\\
a_8 &= \frac{\delta_1^4 + \delta_2^4 + \delta_3^4}{s'^4}\\
a_9 &= \frac{\delta_1^2\delta_2^2\delta_3^2 + 5\delta_1\delta_2\delta_3 s' + 4s'^2}{s'^5}\\
a_{10} &= -\frac{\delta_1^3\delta_2\delta_3 + 2s'\delta_1^2 + \delta_1\delta_2^3\delta_3 + \delta_1\delta_2\delta_3^3 + 2s'\delta_2^2 + 2s'\delta_3^2}{s'^5}\\
a_{11} &= \frac{\delta_1^2\delta_2^2 + \delta_1^2\delta_3^2 + \delta_2^2\delta_3^2}{s'^5}\\
a_{12} &= \frac{s' - \delta_1\delta_2\delta_3}{s'^5}.
\end{aligned} \tag{49}$$

B Application: Image Warping by Moving Least-Squares

We use DCAT within an image warping method based on the Moving Least-Squares (MLS) approach (Schaefer et al., 2006), as illustrated in figure 3. Given the particular use-case of image warping, we work in dimension $d = 2$, though MLS and DCAT both work in greater dimensions, and their general integration can be readily worked out following the derivation given below. The deformation field in MLS is defined by 2D control points and a transformation group. The 2D control points may be handled by a user to specify the desired global deformation sketch. In contrast, the transformation group, usually chosen as the Euclidean, similarity, or affine group, defines the deformation's local behaviour.

We propose to use the $SL_2(\mathbb{R})$ group in order to construct approximate local-area-preserving deformation fields, by integrating DCAT with MLS. Let $P, Q \in \mathbb{R}^{2\times m}$ be the coordinates of the 2D control points in the original and deformed images respectively. In general, the original control points P are evenly distributed within a region of interest or specified by the user, and kept fixed. In contrast, the deformed control points Q, initialised as $Q = P$, may then be interactively edited by the user. The MLS deformation field is obtained point-wise: for an original point with coordinates $p \in \mathbb{R}^2$, a specific affine transformation (A_p, t_p) is produced by involving a weighted estimation of the chosen transformation group, and finding the deformed point coordinates as $q = A_p\, p + t_p$. Area preservation is achieved by solving the DCAT problem of equation (1) in dimension $d = 2$ with P, Q defined as the control points. We use $s = 1$ to preserve the global image scale, though this parameter may be tuned by the user to implement scale control. The localised behaviour is achieved by carefully choosing the weights W as a function of point p. The intuition



(a) Original image (b) Warped image

Figure 3: Image warping: principle. (a) shows the original 600 × 600 pixels image, enclosed with a uniform triangulation with 16 × 16 vertices in gray, and endowed with $m = 16$ control points in blue. (b) shows the warped image, obtained by interactively displacing the control points, applying the MLS principle with DCAT to the mesh vertices and affinely interpolating the transformation within each triangle.

is to weight each control point by a decreasing function of its proximity to point p in the original image. We denote the j-th original control point coordinates as $p_j \in \mathbb{R}^2$, $j \in [1, m]$; in other words, p_j is the j-th column in matrix P. The corresponding weight in W is then chosen as:

$$\omega_j(p) = \frac{1}{(\|p - p_j\| + \epsilon)^{2\alpha}}. \tag{50}$$

This weight function involves two hyperparameters, ϵ and α. Hyperparameter $\epsilon > 0$ defines the maximal weight, which occurs when p coincides with the control point, giving a weight of $w_j(p_j) = 1/\epsilon^{2\alpha}$. It is usually fixed to a small value. Hyperparameter $0 < \alpha \leq 1$ controls the deformation's locality; $\alpha \to 0$ induces a deformation with limited local variations, whereas $\alpha = 1$ induces a more rapidly-varying deformation. It is usually interactively adjusted by the user within the above bounds; the resulting deformation field is generally smooth. However, non-diffeomorphic deformations may occur, manifesting as fold overs in the warped image. These fold-overs are induced by control points being moved towards each other and beyond.

We implement MLS image deformation by generating a uniform triangulation over a grid of $n_g \times n_g$ vertices enclosing the original image domain. Given the current control point and hyperparameter values, the above-described MLS principle is used to transfer each of the n_g^2 vertices to the deformed state. Each triangular facet of the mesh induces an affine transformation which we use to warp the original to the deformed image. We compare the proposed integration of DCAT with MLS, denoted `MLS-DCAT`, with the existing Euclidean, similarity and affine variants, denoted `MLS-EUC`, `MLS-SIM` and `MLS-AFF`, respectively. We use $m = 16$ control points and a triangulation drawn from $n_g \times n_g = 16 \times 16$ vertices laid over the original image shown in figure 3, used for illustration purposes. We use $\epsilon = 10^{-10}$ and $\alpha = 1$ in all experiments. We generate 100 deformation fields by randomly perturbing one of the control points according to a zero-mean Gaussian distribution with a standard deviation of 100 pixels in each coordinate. As the inter-point distance is of the order of $\sqrt{2} \cdot 100/16 \approx 53$ pixels, this means that the perturbed control point will cross over the line joining two of its neighbouring control points in approximately 60% of the cases on average. Figure 4 shows deformation examples for the 4 implemented transformation models. The fold overs are obvious from the visual artefacts they create in the warped image. It is expected that the stiffer the transformation model, the rarer the fold overs. The average proportion of folded-over faces per transformation model is the lowest for `MLS-EUC`, at 0.16%, which reflects this model's natural stiffness. It raises for `MLS-SIM` and `MLS-DCAT`, at 0.33% and 0.30% respectively. Lastly, it is the highest for `MLS-AFF`, at 0.84%, reflecting the model's natural flexibility. It is interesting to compare these statistics with each model's local number of degrees of freedom. Specifically, in dimension $d = 2$, `MLS-EUC` has 3, `MLS-SIM` has 4, `MLS-DCAT` has 5 and `MLS-AFF` has 6. We thus have, in spite of the added flexibility over `MLS-SIM`, that `MLS-DCAT` has a lower rate of fold-over, while, in spite of being also as flexible as `MLS-AFF`, its rate of fold-rate is almost three times lower. This is explained by the signed-area preservation constraint in DCAT, enforcing $\det(A_p) = 1$, striking a balance between deformation amplitude and local-structure preservation.

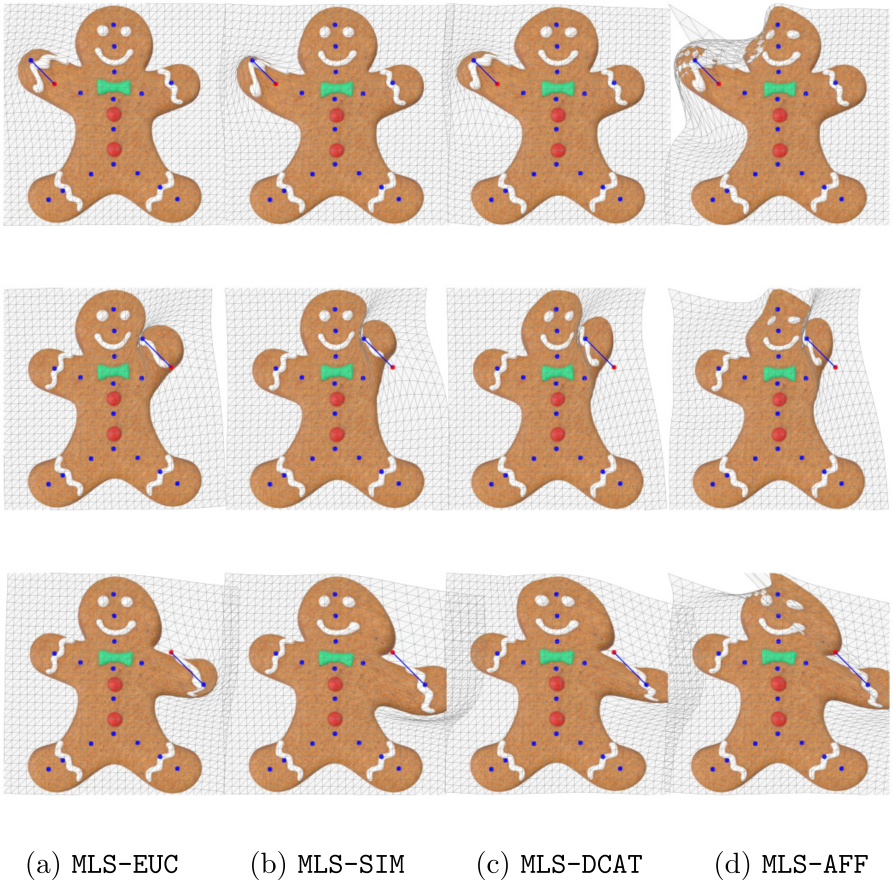


Figure 4: Image warping: examples of warped images with the four tested local models. The deformation field is generated by MLS from the control point in red randomly displaced along the blue line.

C Application: Generalised Procrustes Analysis

Generalised Procrustes Analysis (GPA) is a statistical shape analysis technique, generalising to any number of shapes the original Procrustes analysis designed for two shapes (Gower, 1975; Ten Berge, 1977). GPA finds an 'average' shape which best represents the data shapes by aligning them spatially. The average shape and spatial alignment depend on a user-specified set of variations to compensate for, which are specified by choosing a transformation group, within which the aligning transformations are estimated. GPA is generally used with the Euclidean and the similarity groups; they both eliminate variations in position and orientation, and in scale for the similarity group. GPA assumes that the data shapes are represented by m corresponding points and uses least-squares to measure closeness between the shape points. GPA methods proceed iteratively. They initialise the aligning transformations, typically to the identity transformation of the transformation group, and then alternate two steps: step *i)* is the computation of the average shape, and step *ii)* is the computation of the aligning transformation between each data shape and the average shape. The iterations stop when the change in the average shape is below a threshold chosen as $\epsilon = 10^{-8}$ or when the iteration count reaches 100. A detailed algorithm may be found in (Bartoli et al., 2013).

We compare the alternating GPA algorithm with the classical Euclidean and similarity groups, leading to methods `GPA-EUC` and `GPA-SIM` respectively, with the $SL_3(\mathbb{R})$ group, method `GPA-DCAT`, implemented thanks to the DCAT procedure in algorithm 1 with dimension $d = 3$, determinant $s = 1$ and unit weights. We use AMOS, a public dataset with 70 shapes representing the liver of different individuals reconstructed from CT scans (Ji et al., 2022). Each shape is represented by a triangular 3D surface mesh with a large number of vertices. It is manually annotated with $m = 13$ anatomical points (Labrunie et al., 2024). We use these 13 points to perform GPA and then apply the aligning transformations to the complete vertex sets.

	Individual point correspondence dispersion													
	1	2	3	4	5	6	7	8	9	10	11	12	13	avg
`GPA-EUC`	14.01	11.08	11.31	9.83	9.35	10.62	11.68	10.74	9.67	7.12	11.22	12.45	14.86	11.07
`GPA-SIM`	12.26	9.21	10.35	8.99	8.57	9.56	9.65	9.09	9.09	6.84	10.10	11.39	13.66	9.90
`GPA-DCAT`	11.10	9.96	8.65	7.45	7.21	7.94	8.79	9.19	7.82	5.97	8.50	9.44	11.19	8.71

Table 2: Procrustes analysis: measurement of individual and average point correspondence dispersion.

We use point dispersion to measure the post-GPA residual shape variability. Specifically, we use the aligning transformations to transform all shapes towards the average shape, as in step *ii)* of the GPA algorithm, resulting in multiple occurrences forming a localised point cloud for each of the 13 point correspondences, as illustrated in figure 5. We quantify the dispersion of each point cloud from its sample covariance matrix $C \in \mathbb{R}^{3\times 3}$ as $\text{disp} = \sqrt{\frac{1}{3}\text{trace}(C)}$. The lower the dispersion, the more effective the variation-removal effect of GPA. We give quantitative point dispersion measurements in table 2. We observe that `GPA-EUC` has the higher dispersion in all point correspondences. This is as expected, as the Euclidean group is a subgroup of both the similarity and the $SL_d(\mathbb{R})$ groups, and hence necessarily preserves a higher amount of shape variation. We then observe that `GPA-SIM` generally has a higher dispersion than `GPA-DCAT`, but not always. In particular, `GPA-DCAT` has a slightly higher dispersion for point correspondences 2 and 8. This is explained by the fact that the similarity and the $SL_d(\mathbb{R})$ groups are not subgroups of each other. There are, in fact, complementary: while they both share a Euclidean subgroup, the similarity group performs volume changing, while the $SL_d(\mathbb{R})$ group does not, but adds other variations pertaining to the affine group, such as relative scaling and skewing. As the $SL_d(\mathbb{R})$ group has more degrees of freedom than the similarity group, it is coherent it generally has a lower dispersion in GPA. Figure 6 shows the final result of using the aligning transformations on the complete data shapes. As expected, we observe that the overlap is qualitatively better for `GPA-DCAT`, followed by `GPA-SIM` and `GPA-EUC`.

References

Y. Adagolodjo, R. Trivisonne, N. Haouchine, S. Cotin, and H. Courtecuisse. Silhouette-based pose estimation for deformable organs application to surgical augmented reality. In *International Conference on Intelligent Robots and Systems*, 2017.

C. Arango and A. Bartoli. An optimal triangle projector with prescribed area and orientation, application to position-based dynamics. *Graphical Models*, 118:101–117, November 2021.

A. Bartoli, D. Pizarro, and M. Loog. Stratified generalized procrustes analysis. *International Journal of Computer Vision*, 101(2):227–253, 2013.

J. Bender, M. Müller, and M. Macklin. Position-based simulation methods in computer graphics. In *Eurographics 2015 Tutorials*, 2015.

A. Björck. *Numerical Methods For Least-Squares Problems.* Society For Industrial and Applied Mathematics, 1996.

S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. ACM *Transactions on Graphics*, 33(4), July 2014.

D. A. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Springer-Verlag, 2005.

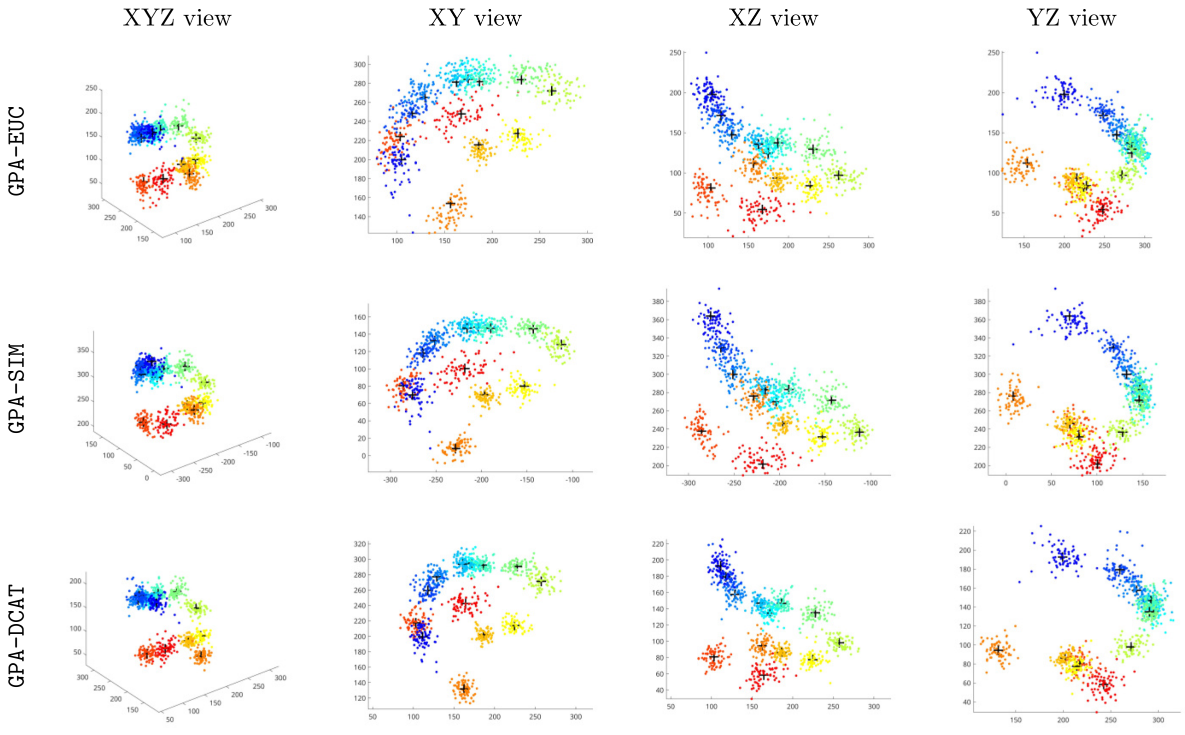


Figure 5: Procrustes analysis: visualisation of the point dispersion. Each of the $m = 13$ point correspondences defines a point cloud shown in a specific colour. The centroid of each point cloud is shown as a black cross.

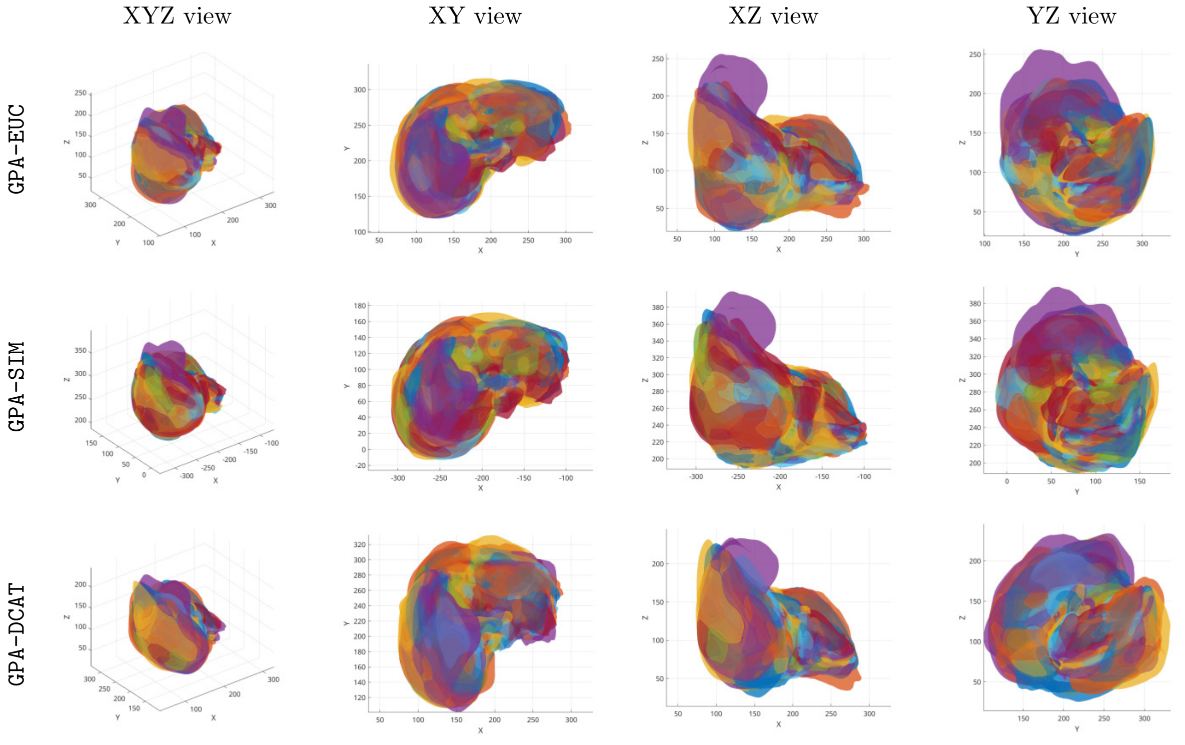


Figure 6: Procrustes analysis: aligned data shapes. Following the run of GPA with the $m = 13$ point correspondences, the complete data shapes are aligned with the computed transformations. Each shape is shown in its own colour.

J. Demmel. Singular value decomposition. In Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. H. van der Vorst, editors, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, chapter 6, pages 135–147. Society for Industrial and Applied Mathematics, 2020.

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

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

Y. Ji, H. Bai, C. Ge, J. Yang, Y. Zhu, R. Zhang, Z. Li, L. Zhang, W. Ma, X. Wan, and P. Luo. AMOS: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. In *Conference on Neural Information Processing Systems*, 2022.

M. H. Kee, K. Um, W. Jeong, and J. Han. Constrained projective dynamics: Real-time simulation of deformable objects with energy-momentum conservation. ACM *Transactions on Graphics*, 40(4), August 2021.

B. Koo, E. Özgür, B. L. Roy, E. Buc, and A. Bartoli. Deformable registration of a preoperative 3D liver volume to a laparoscopy image using contour and shading cues. In *Medical Image Computing and Computer-Assisted Intervention*, 2017.

M. Labrunie, M. Ribeiro, F. Mourthadhoi, C. Tilmant, B. L. Roy, E. Buc, and A. Bartoli. Automatic preoperative 3d model deformable registration in laparoscopic liver resection. *International Journal of Computer Assisted Radiology and Surgery, special issue: selected papers from IPCAI 2022*, 17(8):1429–1436, August 2022.

M. Labrunie, D. Pizarro, C. Tilmant, and A. Bartoli. Generic liver modelling with application to mini-invasive surgery guidance. In *International Conference on Medical Imaging and Computer-Aided Diagnosis*, 2024.

N. Rabbani, L. Calvet, Y. Espinel, B. L. Roy, M. Ribeiro, E. Buc, and A. Bartoli. A methodology and clinical dataset with ground-truth to evaluate registration accuracy quantitatively in computer-assisted laparoscopic liver resection. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 10(4):441–450, June 2022.

S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In SIGGRAPH, 2006.

E. F. Taylor and J. A. Wheeler. *Spacetime Physics*. W. H. Freeman and Company, 1966. Second Edition.

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

T. Terao, K. Ozaki, and T. Ogita. LU-Cholesky QR algorithms for thin QR decomposition. *Parallel Computing*, 92:102571, April 2020.

C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

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.