

Continuous symmetry and shape measures

by Guillem Perarnau-Llobet,¹ Julian Pfeifle² and Jordi Saludes²

1.1 Introduction

A problem in computational chemistry posed during GEMT 2009 at UPC was to find the optimal affine transformation between two point sets $X, Y \subset \mathbb{R}^3$ of n points each that minimizes a certain similarity measure. Given a bijection $\pi: X \rightarrow Y$, the optimal affine transform sending $\pi(Y)$ to X can be computed efficiently by analytic means [3]. The crucial bottleneck encountered in previous work lies with the combinatorial complexity of having to enumerate all $n!$ permutations of these point sets to find the best affine transform.

In this paper, we present an algorithm that *approximately* matches X and Y using affine transformations, and returns the best correspondence between the transformed sets. From this, the best global affine transform can then be computed analytically.

Our strategy is to first translate X and Y so that their respective barycenters lie at the origin, and then scale each set so that the variation of the set of distances to the origin is the same. The only remaining ingredient is to find a rotation $R \in \text{SO}(3)$ that makes $R(X)$ and Y agree “as far as possible”.

1.1.1 Error measures

We will take the sum of squared distances of matching points. Another viable option would be Hausdorff distance [1].

¹Departament de Matemàtica Aplicada 4, UPC

²Departament de Matemàtica Aplicada 2, UPC

1.2 The Cayley chart of $\text{SO}(3)$

The Lie group $\text{SO}(3)$ of orthogonal 3×3 matrices with determinant 1 has many different charts. The most important for us is that given by the *Cayley transform*

$$\begin{aligned} \mathcal{C} : \text{so}(3) &\longrightarrow \text{SO}(3) \setminus M \\ A &\longmapsto (I - A)(I + A)^{-1}. \end{aligned}$$

It establishes a bijection between skew-symmetric matrices

$$A = \text{sk}(x, y, z) = \begin{pmatrix} 0 & x & -y \\ -x & 0 & z \\ y & -z & 0 \end{pmatrix} \in \text{so}(3)$$

with $x, y, z \in \mathbb{R}$ and the set $\text{SO}(3) \setminus M$, where M denotes the set of rotation matrices that have -1 as an eigenvalue. Specifically, it takes A to

$$\mathcal{C}(\text{sk}(x, y, z)) = \frac{1}{\Delta} \begin{pmatrix} 1 + x^2 - y^2 - z^2 & 2xy - 2z & 2(y + xz) \\ 2(xy + z) & 1 - x^2 + y^2 - z^2 & -2x + 2yz \\ -2y + 2xz & 2(x + yz) & 1 - x^2 - y^2 + z^2 \end{pmatrix},$$

where $\Delta = 1 + x^2 + y^2 + z^2$.

The inverse map is given by the same expression,

$$\mathcal{C}^{-1}(Q) = (I - Q)(I + Q)^{-1} \quad \text{for } Q \in \text{SO}(3) \setminus M.$$

We need to find the set of rotation matrices that map a point with spherical coordinates (θ_1, φ_1) in the 2-dimensional sphere S^2 to the point with spherical coordinates (θ_2, φ_2) . Elementary calculations yield the following result:

Proposition 1.2.1. *The inverse image under the composite map $\mathcal{C} \circ \text{sk}$ of the set of rotations that send $u = (\theta_1, \varphi_1)$ to $v = (\theta_2, \varphi_2)$ is the affine line ℓ in $(\mathbb{R}^3, (x, y, z))$ given by*

$$\begin{aligned} x &= \frac{\cos(\varphi_2) - \cos(\varphi_1) + y(\cos(\theta_2) \sin(\varphi_2) + \cos(\theta_1) \sin(\varphi_1))}{\sin(\varphi_2) \sin(\theta_2) + \sin(\varphi_1) \sin(\theta_1)}, \\ z &= \frac{y(\cos(\varphi_2) + \cos(\varphi_1)) - \cos(\theta_2) \sin(\varphi_2) + \cos(\theta_1) \sin(\varphi_1)}{\sin(\varphi_2) \sin(\theta_2) + \sin(\varphi_1) \sin(\theta_1)}. \end{aligned}$$

If $u = (u_1, u_2, u_3)$ and $v = (v_1, v_2, v_3)$ are the Cartesian coordinates of u , respectively v , then a point p on ℓ and a direction vector a for ℓ are given by

$$p = \left(\frac{-u_3 + v_3}{u_2 + v_2}, 0, \frac{u_1 - v_1}{u_2 + v_2} \right), \quad a = (u_1 + v_1, u_2 + v_2, u_3 + v_3).$$

1.2.1 Computing rotations

We will try to match a given triangle in the reference point set to every suitable triangle in the problem set using a rotation. From the above, it is clear that finding the optimal rotation that achieves this corresponds to intersecting the three lines $q_i + t_i v_i$ for $i = 1, 2, 3$ in the Cayley parametrization space. If the triangles in question are congruent, these three lines will meet at a single point, so the problem is overdetermined; if the triangles are not congruent, however, the three lines will not intersect at all. We therefore choose to solve the problem of minimizing the sum of squared distances

$$D = \sum_{i,j}^3 \|q_i - q_j + t_i v_i - t_j v_j\|^2.$$

By computing the gradient of D with respect to the unknowns t_i , we obtain the equivalent system of linear equations

$$\sum_{j \neq i} (q_j - q_i) \cdot v_i = 2t_i \|v_i\|^2 - \sum_{j \neq i} t_j v_i \cdot v_j \quad \text{for } i = 1, 2, 3,$$

which expressed in matrix form reads as follows:

$$\begin{pmatrix} 2\|v_1\|^2 & -v_1 \cdot v_2 & -v_1 \cdot v_3 \\ -v_2 \cdot v_1 & 2\|v_2\|^2 & -v_2 \cdot v_3 \\ -v_3 \cdot v_1 & -v_3 \cdot v_2 & 2\|v_3\|^2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} \sum_{j \neq 1} (q_j - q_1) \cdot v_1 \\ \sum_{j \neq 2} (q_j - q_2) \cdot v_2 \\ \sum_{j \neq 3} (q_j - q_3) \cdot v_3 \end{pmatrix}.$$

As a candidate for the optimal rotation, we take the one corresponding to the barycenter of the solution points: $\frac{1}{3} \sum_{i=1}^3 q_i + t_i v_i$. Alternatively, we could consider the minimization problem

$$D' = \sum_{i=1}^3 \|q_i + t_i v_i - p\|^2$$

with unknowns $t_i \in \mathbb{R}$ and $p \in \mathbb{R}^3$.

1.3 Approximate affine point matching

1.3.1 Overview of the algorithm

The input data are two ordered sets $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n) \subset \mathbb{R}^3$ of n points each. We want to compute a permutation $\pi \in S_n$ such that

the ordered set $Y_\pi = (y_{\pi(1)}, \dots, y_{\pi(n)})$ approximately minimizes the *shape measure* $S(X, Y) = \min_{\pi \in S_n} S_\pi(X, Y)$, where

$$S_\pi(X, Y) = \min_{f \text{ affine}} \frac{\sum_{i=1}^n \|x_i - f(y_{\pi(i)})\|^2}{\sum_{i=1}^n \|x_i - \beta\|^2}.$$

Here $\beta = \frac{1}{n} \sum_{i=1}^n x_i$ is the barycenter of X , and we take the minimum over *all* affine transformations of \mathbb{R}^n . We find a permutation that approximates $S_\pi(X, Y)$ using only a finite number of such transformations.

The first steps are to translate the barycenters of X and Y to the origin, and to scale both sets so that the variances of their distances to the origin equal some fixed value. We retain the names X and Y for these translated and scaled sets. After this, we need to optimize over all rotations.

To any rotation $R \in \text{SO}(3)$ we associate the map $\pi = \pi(R) : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ that assigns to each point $x_i \in X$ the point $y'_{\pi(i)} \in Y' = R(Y)$ closest to it in the Euclidean norm. In favorable cases, for example when Y is an affine image of a slight perturbation of X , this map π is actually a permutation of $\{1, \dots, n\}$. The optimal affine transform that maps X to Y_π can then be found by analytical means.

Denote the set of triangles formed by points in X and Y by \mathcal{T}_X and \mathcal{T}_Y , respectively. To find a good set of candidate rotations, we first choose a certain (relatively small) subset $\mathcal{T}'_X \subset \mathcal{T}_X$ of the triangles in X . For each such triangle $T_X \in \mathcal{T}'_X$, we iterate over *all* triangles $T_Y \in \mathcal{T}_Y$, and for each pair (T_X, T_Y) we find the rotation $R \in \text{SO}(3)$ that most closely maps T_X onto T_Y using the methods in the preceding section. We then apply R to the entire set X , find the corresponding optimal permutation $\pi(R)$, and calculate the associated shape measure $S_{\pi(R)}(X, Y)$. Finally, we return the permutation corresponding to the best rotation among all those seen throughout the process.

1.3.2 Implementation details

Choosing \mathcal{T}'

In general, the centered and scaled sets X and Y will not lie on a sphere. Thus, two points $x_i \in X$ and $y_j \in Y$ will generally have different norms. If this is true, it makes little sense to try to rotate x_i into y_j . Reciprocally, if X and Y are almost affine images of each other, it stands to reason that the distribution of the norms of their elements will be similar.

Algorithm 1 Pseudocode of the matching algorithm

```

1: procedure MATCHING( $X, Y$ )
2:    $X$ : the reference points
3:    $Y$ : the problem points
4:   Set  $X := \text{scale}(\text{center}(X))$ 
5:   Set  $Y := \text{scale}(\text{center}(Y))$ 
6:   global_error := 0
7:    $x_1 := \text{closest\_to\_origin}(X)$ 
8:    $x_2 := \text{furthest\_to\_origin}(X)$ 
9:   for all triangles  $T_X = \{x_1, x_2, x\} \in \mathcal{T}_X$  do
10:    for all examinable triangles  $T_Y = \{y_1, y_2, y_3\}$ , where  $y_i \in Y$  do
11:       $R := \text{optimal\_rotation}(T_X, T_Y)$ 
12:       $X_R := \text{rotate}(X, R)$ 
13:      if matching_error( $X_R, Y$ ) < global_error then
14:        actualize the matching and its error
15:      end if
16:    end for
17:  end for
18:  Set  $\hat{X} := \text{scale}(\text{center}(\text{specular}(X)))$ 
19:   $\hat{x}_1 := \text{closest\_to\_origin}(\hat{X})$ 
20:   $\hat{x}_2 := \text{furthest\_to\_origin}(\hat{X})$ 
21:  for all triangles  $T_{\hat{X}} = \{\hat{x}_1, \hat{x}_2, \hat{x}\} \in \mathcal{T}_{\hat{X}}$  do
22:    for all examinable triangles  $T_Y = \{y_1, y_2, y_3\}$ , where  $y_i \in Y$  do
23:       $R := \text{optimal\_rotation}(T_{\hat{X}}, T_Y)$ 
24:       $X_R := \text{rotate}(X, R)$ 
25:      if matching_error( $\hat{X}_R, Y$ ) < global_error then
26:        actualize the matching and its error
27:      end if
28:    end for
29:  end for
30: end procedure

```

In the hope of rapidly and accurately capturing the shape of Y , we therefore choose the points x_{\min} and x_{\max} of minimal and maximal norm to always form part of the initial triangle T_X . This leaves us with a linear number of initial triangles:

$$\mathcal{T}_X = \{ \text{conv}\{x_{\min}, x_{\max}, x\} : x \neq x_{\min}, x_{\max} \}.$$

Sometimes rotation does not suffice because the X and Y have different orientations. Hence we may define $\hat{X} = \sigma(X)$, where σ is some reflection, for example that with respect to the plane $\{x = 0\}$. Analogously, we define $\mathcal{T}_{\hat{X}} = \sigma(\mathcal{T}_X)$. Note that $\hat{x}_{\min} = \sigma(x_{\min})$ and $\hat{x}_{\max} = \sigma(x_{\max})$.

Finding the error given R

To calculate the error induced by a rotation R , we must compute the map $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$. We use a k - d -tree built from X to rapidly query the closest corresponding rotated point. This gives us an injective map that is not necessarily exhaustive; however, this has always been the case in the experiments we have conducted. Note that we build the k - d -tree on the fixed reference set, so that we only have to execute this preprocessing once.

Sorting pairs of points by difference in norm

Another crucial optimization to find the optimal rotation is the following. We calculate the norms of all points in \mathcal{T}_X and Y , and sort the list $(\|x\| - \|y\| : x \in \mathcal{T}_X, y \in Y)$ of absolute values of their differences by size. We then use the two triangles formed by the first three pairs of points from this sorted list to calculate the first candidate rotation R . Intuitively, this makes sense because we expect these triangles to be quite similar. We then proceed with other candidate triangles from the beginning of the list. One must take a little care to check that each triple of the selected pairs really consists of six distinct points.

Due to this optimization, for each triangle in \mathcal{T}_X we only examine certain triangles in \mathcal{T}_Y (the first according to this sorted list), and this improves the execution time.

1.4 Results

We experimentally evaluate the efficiency of our algorithm in terms of time complexity and quality of the solution found. We test it in the following examples:

- A 7-vertex polyhedron with a central vertex.
- A cluster Au_{28} consisting of 28 gold atoms.
- Instances of a scalable artificial dataset. In order to be able to experiment with large datasets, we have implemented a program that outputs an arbitrarily large point cloud and a perturbation of it, and allows the amount of perturbation to be tuned.

Each instance is accompanied by a perturbed version, which we then try to match.

1.4.1 7-vertex polyhedron with a central vertex

This dataset consists of a polyhedron with 7 vertices on its convex hull, along with another point at the barycenter. Exhaustive enumeration confirms the permutation output by our algorithm to be the optimal one.

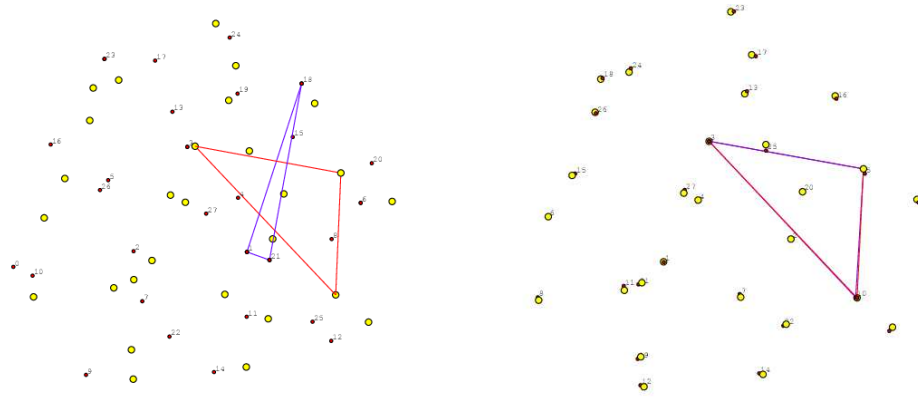
We do not apply an optimal analytical minimization of the distances between matched points; hence, we obtain $S(Q, P) \leq 0.93518$ instead of the optimal $S(Q, P) = 0.47764$. This shows that, despite computing the optimal permutation, we overestimate $S(Q, P)$.

The algorithm spends approximately two seconds on this example.

1.4.2 Au_{28}

For this 28-point instance, it is computationally out of the question to enumerate all $28! \approx 3 \times 10^{29}$ permutations. Other heuristic methods [2] have obtained a permutation of the nodes in Au_{28} that lead to the bound $S(Q, P) \leq 1.69182$. This heuristic consists of finding an optimal plane to apply a specular symmetry, assuming that no rotation is needed.

Here the specular symmetry approach is very important, since the two datasets do not have the same orientation. Our algorithm finds a substantially different permutation, leading to an upper bound for the symmetry measure of $S(Q, P) \leq 0.23426$, which improves the former. This solution is found in 20 seconds.



(a) Au_{28} and ideal polyhedra before rotation. (b) Au_{28} and ideal polyhedra after rotation.

Figure 1.1: The Au_{28} dataset and its perturbed version before and after rotation, but *after* applying the mirror symmetry. In both images, we have marked the triangles that select the optimal rotation.

1.4.3 Scalable artificial dataset

Finally, to test the real time complexity and the quality of the solution when both the size increases and the quantity of perturbation varies, we use our artificial dataset generator. The time spent by the algorithm depending on the size of the point cloud is shown in Figure 1.2. We also show our upper bound on the symmetry measure compared with the real one given a fixed size and varying the perturbation of the points.

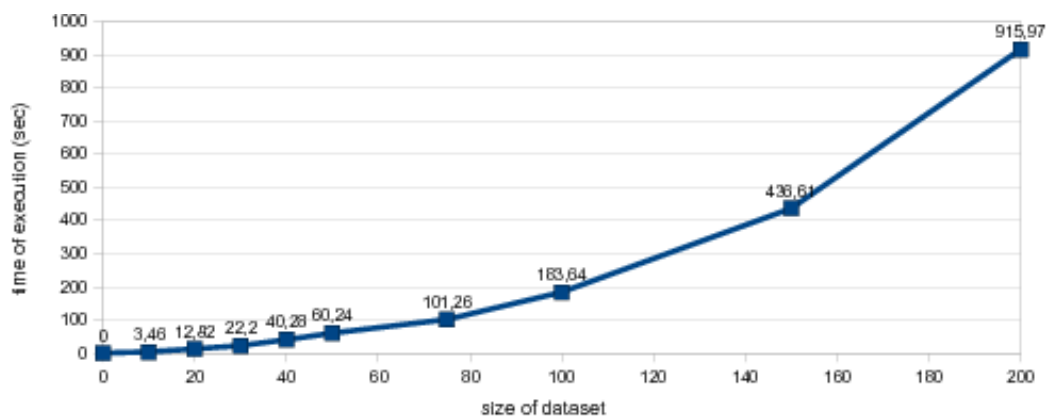


Figure 1.2: Execution time of the matching algorithm in an artificial set of points.

1.5 Other ideas

1.5.1 Local minimization techniques

Another possibility is to define a smooth function such as the one given by an attractive potential between the reference points $P = \{p_i\}$ and the problem points $Q = \{q_i\}$,

$$V(P, Q) = \sum_{i,j} \phi(\|p_i - q_j\|),$$

where $\phi(r) = -r^{-\alpha}$ for $\alpha > 0$. Then, given a rotation R and a local chart containing R with coordinates x, y, z , we will try a steepest descent method on $(x, y, z) \mapsto V(R(x, y, z)P, Q)$. In this way R is changed slightly to get a better match.

To avoid falling into local minima, we should grid the group of rotations and take the elements of the grid as initial values for the steepest descent method.

Bibliography

- [1] O. AICHHOLZER, H. ALT, AND G. ROTE, *Matching shapes with a reference point.*, Int. J. Comput. Geom. Appl., 7 (1997), pp. 349–363.
- [2] P. ALEMANY, private communication.
- [3] M. PINSKY, C. DRYZUN, D. CASANOVA, P. ALEMANY, AND D. AVNIR, *Analytical methods for calculating continuous symmetry measures and the chirality measure*, J. Comput. Chem., 29 (2008), pp. 2712–21.