Congruence Testing of Point Sets in Three and Four Dimensions

Results and Techniques

Günter Rote[1]

Institut für Informatik, Freie Universität Berlin, rote@inf.fu-berlin.de

Abstract. I will survey algorithms for testing whether two point sets are congruent, that is, equal up to an Euclidean isometry. I will introduce the important techniques for congruence testing, namely dimension reduction and pruning, or more generally, condensation. I will illustrate these techniques on the three-dimensional version of the problem, and indicate how they lead for the first time to an algorithm for four dimensions with near-linear running time (joint work with Heuna Kim). On the way, we will encounter some beautiful and symmetric mathematical structures, like the regular polytopes, and Hopf-fibrations of the three-dimensional sphere in four dimensions.

1 Problem Statement

Given two n-point sets $A, B \subset \mathbb{R}^d$, we want to decide whether there is a translation vector t and an orthogonal matrix R such that $RA + t := \{ Ra + t \mid a \in A \}$ equals B, that is, A and B are *congruent*. Congruence asks whether two objects are the same up to Euclidean transformations, or in other words, whether they are considered equal from a geometric viewpoint. Congruence is therefore one of the fundamental basic notions.

The translation vector t can be easily eliminated from the problem by initially translating the two sets A and B such that their centers of gravity lie at the origin O.

If we do not restrict the dimension d, congruence becomes equivalent to graph isomorphism: a given graph $G = (V, E)$ with n vertices $v_1, \ldots, v_n$ can be represented by $n + |E|$ points in n dimensions. We simply take the n standard unit vectors $e_1, \ldots, e_n$ and add a point $(e_i + e_j)/2$ for each edge $v_iv_j \in E$. Then two graphs are isomorphic if and only if their corresponding point sets are congruent.

We thus restrict our attention to small dimensions. In two and three dimensions, algorithms with a running time of $O(n \log n)$ have been known. We review some of these algorithms, because their techniques are also important for higher dimensions.

The Computational Model: Exact Real Arithmetic. We use the Real Random-Access Machine (Real-RAM) model, as is common in Computational Geometry.

We assume that we can compute arithmetic operations and square roots of real numbers exactly in constant time. The reason for this choice is not so much convenience, but the range of possible input instances. With rational inputs, for example, one cannot even realize a regular pentagon. Thus, the difficult problem instances, which are the symmetric ones, as we will see, would disappear.

It makes sense to ask for approximate congruence within some given tolerance ε. This problem is, however, NP-hard already in two dimensions (Iwanowski 1991). It becomes polynomial when the input points are sufficiently separated in relation to ε, and thus there is hope to solve the approximate congruence problem in higher dimensions, under suitable assumptions and at least in an approximate sense. This is left for future work.

2 Two Dimensions

In the plane, congruence can be tested by string-matching techniques (Manacher 1976). We sort the points clockwise around the origin, in $O(n \log n)$ time, and represent the point set as a cyclic string alternating between n distances from the origin and n angular distances between successive points. Two n-point sets A and B are then congruent if and only if their string representations α and β are cyclic shifts of each other. This is equivalent to asking whether α is a substring of $\beta\beta$, and it can be tested in linear time.

This idea can be extended to symmetry detection for a single set A: We find the lexicographically smallest cyclic reordering of the string. The starting point of this string, together with the cyclic shifts which yield the same string, gives rise to a set of p equidistant rays starting from the origin, which we call the *canonical axes*. Then the set A has a rotational symmetry group of order p, consisting of all rotations that leave the set of canonical axes invariant.

3 Three Dimensions

For testing congruence in space, there are several algorithms, which use different tools (Sugihara 1984, Atkinson 1987, Alt, Mehlhorn, Wagener, and Welzl 1988). We describe a variation which is very simple and illustrates the principal techniques that are used in this area: *dimension reduction*, *pruning*, and *condensation*.

Pruning and condensation tries to successively reduce A to a smaller and smaller point set A' while not losing any symmetries that A might have. Initially, we set $A' := A$. We compute the convex hull $H(A')$ of A' in $O(|A'| \log |A'|)$ time. Let $\bar{A}'$ denote the set of vertices of the polytope $H(A')$. We classify the points of $\bar{A}'$ by degree in the graph of $H(A')$. In case there are at least two different degrees in the graph, we replace A' by the smallest degree class in $\bar{A}'$ and repeat the convex-hull computation. In each iteration, the size of A' is reduced to half or less. We simultaneously carry out all steps for the set B. If at any stage, we notice an obvious difference between A' and B', for example, if $|A'| \neq |B'|$, we conclude that A and B are not congruent, and we terminate.

This pruning loop ends when all vertices in $H(A')$, and also in $H(B')$, have the same degree. At first glance, this procedure looks dangerous because we have thrown away points (including all points interior to the hulls of A and B) and have thereby *thrown away information*: The sets A' and B' might be congruent, whereas the original sets A and B are not. However, the prime goal of successive pruning steps is to eventually reduce the points sets to some sets A' and B' which are so small that we can afford to try all possibilities of mapping a fixed chosen point $u_0 \in A'$ to some point $v \in B'$. This is done as follows:

Once we have picked the point v, we can *reduce the dimension* of the problem by one: we choose some rotation R that brings u_0 to v. We denote by P the plane perpendicular to the axis through $Ru_0 = v$, and we project the sets RA and B onto P. (Here we must take the *original* sets A and B again.) To each projected point, we attach the signed distance from P as a label. We then look for two-dimensional congruences in P, but for *labeled* point sets. The labeling information can be easily incorporated into the algorithm of Sect. 2.

Thus, when $|A'| = |B'|$ is small, we can finish the problem by $|A'|$ instances of two-dimensional congruence in $O(|A'|n \log n)$ time.

Let us now see how we continue when our pruning process gets stuck. We will describe the steps only for the set A', but the reader has to keep in mind that they are carried out for the set B' in parallel. If the convex hull $H(A')$ is one-dimensional or two-dimensional, then we have found an axis or a plane with a corresponding axis or plane in $H(B')$. This allows us to reduce the question to one or two two-dimensional problems, as described above.

We are left with the case that $H(A')$ is a three-dimensional polytope. By pruning, we can assume that all vertices of the graph of $H(A')$ have the same degree d. By Euler's formula, d can be 3, 4, or 5. Euler's formula also yields the number of faces F in terms of the number n' of vertices of $H(A')$: $|F| = (d-2)/2 \cdot n' + 2 \le \frac{3}{2}n' + 2$. We now try to prune the *faces* by face degrees. If there are at least two different face degrees, the smallest degree class F' of faces has at most $\frac{3}{4}n' + 1$ elements. This number is smaller than n' unless $n' = 4$ and $H(A')$ is a tetrahedron. We compute the centers of gravity of the faces in F', and replace A' by the set of these centers. We call this procedure a *condensation*. Like pruning, it reduces A' to a smaller set, but in contrast to pruning, the smaller set is not necessarily a subset of A'.

With the new condensed set A' we restart the whole procedure from scratch, beginning with the convex hull computation. The only case where neither condensation, nor pruning, nor dimension reduction is possible is a convex polytope $H(A')$ in which all vertices and all faces have the same degree. Such a polytope must have the combinatorics of one of the five regular polytopes (Platonic solids): the tetrahedron, the octahedron, the icosahedron, the cube, or the dodecahedron. We know therefore that $|A'| \le 20$, and we can resort to dimension reduction, which leads to at most 20 two-dimensional instances.

In all the above-mentioned pruning and reduction steps, we must avoid that the reduced set A' contains only the origin. When such a case would arise, we artificially select a different class of vertices or faces.

4 Pruning and Condensation

Pruning is very versatile: we can use any criterion of points that we can think of, as long as it is not too expensive to compute. For example, in our algorithm for four dimensions, we will build the closest-pair graph G, which connects all pairs of points of A' whose distance equals the smallest inter-point distance in the set, and try to prune by degree in this graph. If, however, all vertices happen to have degree 1 in G, thus forming a perfect matching of A', we condense A' to the set of midpoints of the matching edges.

The power of the pruning technique is that we can concentrate on those cases where pruning fails. These instances are highly symmetric and regular, and we will capitalize on this regularity to extract structures from the point set that allow us to proceed.

Formally, a condensation procedure is a mapping F that maps a set A to a set $A' = F(A)$. This mapping must be *equivariant* under rotations:

$$R \cdot F(A) = F(R \cdot A), \text{ for all rotations } R$$

A pruning procedure is the special case where $F(A) \subseteq A$. We say that condensation is *successful* if $F(A)$ is smaller than A and $F(A)$ is not the empty set or just the origin. We will be able to ensure a reduction by a constant factor for successful condensation steps, and thus we need not worry about the time for iterating the condensation, because the size of A' decreases at least geometrically.

5 The Three-Dimensional Point Groups

We have seen that congruence testing is closely connected to symmetry: "Random" point sets have no symmetries and are easy to check for congruence. The hard cases are the symmetric ones. It is therefore no surprise that congruence testing algorithms can tell us something about the symmetry groups of point sets.

In Sect. 3, we have stopped condensation as soon as we reached the combinatorial structure of a Platonic solid. By further condensation, based the edge lengths, we can achieve that the only remaining cases must also have the *geometry* of a Platonic solid, see Algorithm K in Kim and Rote (2016) for details. From this we can conclude the following theorem.

Theorem 1. *The symmetry group of a finite three-dimensional set of points is either*

1. *the symmetry group of one of the five Platonic solids,*
2. *the symmetry group of a prism over a regular polygon,*
3. *or a subgroup of one of the above groups.* □

These groups are the discrete subgroups of the orthogonal group $O(3)$ of 3×3 orthogonal matrices, and they are called the *three-dimensional point groups.* Case 2 covers the *reducible groups* (and their subgroups), those groups that are direct products of lower-dimensional point groups. They come from the case when our algorithm used dimension reduction. Theorem 1 is not very explicit, and quite redundant: The octahedron and the cube are dual to each other and have the same symmetries, and so do the dodecahedron and the icosahedron. The tetrahedral group is contained both in the group of the cube and of the dodecahedron. With some work, the explicit list of groups can be worked out from this theorem. However, the resulting classification of three-dimensional point groups was already known in the 19th century (Hessel's Theorem). We will mention potential extensions to four dimensions in Sect. 8.

6 General Dimensions

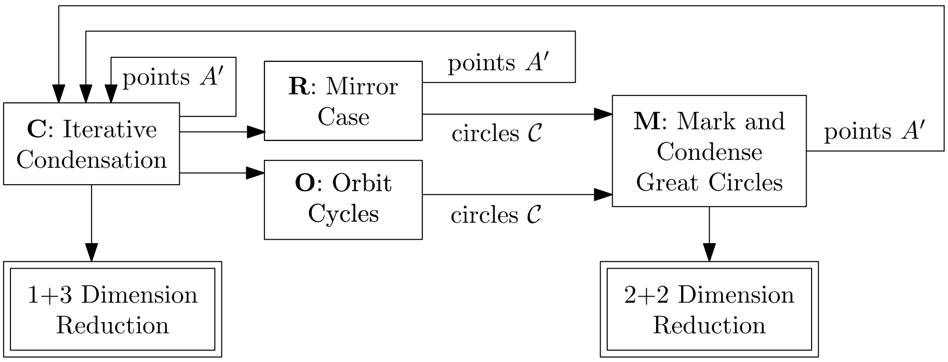
The best algorithms for general dimension d are a deterministic algorithm of Brass and Knauer (2002) and a randomized algorithm of Akutsu (1998). They reduce the dimensionality d of the problem by three, respectively four dimensions at a time, and achieve running times of $O(n^{\lceil d/3 \rceil} \log n)$ and $O(n^{\lfloor d/2 \rfloor/2} \log n)$, respectively, for high enough dimensions.

7 Four Dimensions

We have recently managed to solve congruence testing in four dimensions in optimal $O(n \log n)$ time.

Theorem 2. *Given two sets A and B of n points in four dimensions, it can be decided in $O(n \log n)$ time and $O(n)$ space whether A and B are congruent.*

The algorithm is based on condensation and dimension reduction, but the details are quite involved, see Kim and Rote (2016). We can therefore give only rough overview, referring to the following flowchart, and glossing over many details.



7.1 Iterative Pruning and Condensation Using the Closest-Pair Graph (Algorithm C)

After pruning by distance from the origin, we can assume that A lies on the three-dimensional sphere $\mathbb{S}^3 \subset \mathbb{R}^4$. As in Atkinson (1987), we compute the closest distance $\delta = \min\{\, \|a - a'\| : a, a' \in A, a \neq a' \,\}$ and the *closest-pair graph* G on the vertex set A, which connects all pairs of points whose distance is δ. The vertex degrees in H are bounded by the *kissing number* $K_3 = 12$, the maximum number of equal balls with disjoint interiors that can simultaneously touch a ball of the same size on $\mathbb{S}^3$. The closest-pair graph can be computed by divide and conquer in $O(n \log n)$ time in any fixed dimension (Bentley and Shamos 1976).

Now we start an iterative pruning and condensation process on G, first based on vertex degrees, and working its way up to higher and higher orders of regularity. In the end, we will have pruned G to such a degree that all *directed edge figures*, consisting of some edge uv and all adjacent edges, are congruent. This allows us to conclude that copies of a certain pattern can be found "everywhere" in G. This pattern is a path $t_0u_0v_0w_0$ with the property that its three edges t_0u_0, u_0v_0, and v_0w_0 have the same length δ and the two angles $t_0u_0v_0$ and $u_0v_0w_0$ are equal. In G, we can then define a nonempty set S of paths $aa'a''$ with the following property.

> For every path $a_1a_2a_3 \in S$, there is a (unique) edge $a_3a_4 \in G$ such that $a_2a_3a_4 \in S$ and $a_1a_2a_3a_4$ is congruent to $t_0u_0v_0w_0$.

7.2 Generating Orbit Cycles (Algorithm O)

By repeatedly applying this property, we can conclude:

> For every triple $a_1a_2a_3 \in S$, there is a unique cyclic sequence $a_1a_2 \dots a_\ell$ such that $a_ia_{i+1}a_{i+2}a_{i+3}$ is congruent to $t_0u_0v_0w_0$ for all i. (Indices are taken modulo l.)
> Moreover, there is a rotation matrix R such that $a_{i+1} = Ra_i$. In other words, $a_1a_2 \dots a_\ell$ is the orbit of a_1 under the rotation R.

We call such a cyclic sequence an *orbit cycle*. If the points A would live in $\mathbb{R}^3$, the geometric situation is easy to imagine: If the points $t_0u_0v_0w_0$ lie in a plane, then the orbit cycle lies on a circle. Otherwise, they form an infinite helix that winds around an axis. This intuition is not misleading: on $\mathbb{S}^3$, the situation is the same, except that the axis of the helix is a great circle instead of a line.

The last case it the most interesting case for us: If the points $t_0u_0v_0w_0$ do not lie in a plane, we can extract the axis circle from each orbit cycle. We will then work with the set $\mathcal{C}$ of these circles.

7.3 Marking and Condensation of Great Circles (Algorithm M)

We are given a set $\mathcal{C}$ of great circles in $\mathbb{S}^3$. We will treat these circles as objects in their own right, independent of the point set A from which they came.

The Distance between Circles. We start by computing the closest-pair graph on $\mathcal{C}$. To to this, we have to define a distance between great circles. We do this by embedding them in the 5-sphere $\mathbb{S}^5 \subset \mathbb{R}^6$. Great circles in the 3-sphere can be equivalently regarded as 2-dimensional planes through the origin in 4-space, and we can use Plücker coordinates to represent them. (Planes in 4-space can be equivalently regarded as lines in (projective) 3-space, and this is the most familiar type of Plücker coordinates.) The Plücker coordinates are a 6-tuple of numbers in projective 6-space. We normalize them and represent each circle as a pair of antipodal points on $\mathbb{S}^5$, and define the *Plücker distance* between two circles as the smallest distance between the four representative points. This distance is a geometric invariant: In a different coordinate system, a plane will have different Plücker coordinates, but Plücker distances are unchanged.

Other distances have been considered in the literature. Conway, Hardin, and Sloane (1996) have tried to pack lines, planes, etc. in Grassmannian spaces, using the *chordal distance* (which comes from representing a plane as a symmetric 4×4 projection matrix) and the *geodesic distance* on the Plücker surface. For our case, the Plücker distance gives the embedding of lowest dimension and is therefore preferable.

The closest-pair graph $G(\mathcal{C})$ is thus computed in 6 dimensions. The number of neighbors is bounded by the kissing number K_5 in 5 dimensions, which is known to be bounded by 44.

We now look at each pair C, D of adjacent circles in $G(\mathcal{C})$. When projecting D on the plane of C, the image will generically be an ellipse D'. We use the major axis of D' to *mark* two points on C. Similarly, we project C to the plane of D and generate two markers on D. Repeating this for all edges of $G(\mathcal{C})$ produces at most $2K_5 \le 88$ markers on each circle of $\mathcal{C}$. These markers form a new set of points A', and we start the whole algorithm from scratch with this set of points.

We argue that the new set A' is smaller than the original set A from which the orbit cycles are generated. We know that every point of A can belong only to a bounded number of orbit cycles, by the degree constraint in $G(A)$. If all orbit cycles are *long* enough, meaning that they contain sufficiently many points, we can therefore guarantee that the number of orbit cycles is small, say $|\mathcal{C}| \le |A|/200$, and then $|A'| \le 88 \cdot |\mathcal{C}|$ will be a successful condensation of A. If the orbit cycles are short, it means that the closest distance δ must be longer than some threshold δ_0. Then, by a straightforward packing argument on $\mathbb{S}^3$, the size of A is bounded by a constant, and we can "trivially" solve the problem by dimension reduction.

Isoclinic Circles and Hopf Bundles. The above procedure fails to generate markers if all projected ellipses turn out to be circles. Such planes C, D are called *isoclinic.* They come in two variations, *left-isoclinic* and *right-isoclinic.* It turns out that being isoclinic imposes a strong structure on the involved circles. We formulate their properties for right-isoclinic pairs; analogous statements hold for left-isoclinic pairs.

Proposition 3. *1. The relation of being right-isoclinic is transitive (as well as reflexive and symmetric). An equivalence class is called a* right Hopf bundle*.*

2. For each right Hopf bundle, there is a right Hopf map *h that maps the circles of this bundle to points on $\mathbb{S}^2$.*

3. By this map, two isoclinic circles with Plücker distance $\sqrt{2}\sin\alpha$ are mapped to points at angular distance 2α on the "Hopf sphere" $\mathbb{S}^2$.

4. A circle can have at most $K_2 = 5$ *closest neighbors on the Plücker sphere* $\mathbb{S}^5$ *that are right-isoclinic.*

The right Hopf map in Property 2 is obtained as follows (Hopf 1931, §5): Choose a positively oriented coordinate system x_1, y_1, x_2, y_2 for which some circle C_0 of the bundle lies in the x_1y_1-plane. Then the map $h\colon \mathbb{S}^3 \to \mathbb{S}^2$ defined by

$$h(x_1, y_1, x_2, y_2) = \big(2(x_1y_2 - y_1x_2),\ 2(x_1x_2 + y_1y_2),\ 1 - 2(x_2^2 + y_2^2)\big)$$

maps all points on a circle of the bundle to the same point on $\mathbb{S}^2$. A different choice of C_0 would lead to a different map, but by Property 3, the images are related by an isometry of $\mathbb{S}^2$. The constant $K_2 = 5$ in Property 4 is the kissing number on the 2-sphere. Property 4 is a direct consequence of Properties 2 and 3.

We use Proposition 3 in the following way: If all pairs of circles in a component of the closest-pair graph $G(\mathcal{C})$ are right-isoclinic, we know that they must belong to a common Hopf bundle. We then use a condensation procedure on the Hopf sphere, similar to the one described in Sect. 3, to condense the set of circles, and repeat the construction of the closest-pair graph.

If a circle C has both a left-isoclinic neighbor D and a right-isoclinic neighbor D', we conclude by Property 1 that D and D' cannot be isoclinic. We can therefore mark points on D and D'.

To summarize, we repeatedly condense the set $\mathcal{C}$ of circles until we can mark some points A' on them, or until the number of circles in $\mathcal{C}$ gets smaller than some threshold. In the latter case, we apply 2+2 Dimension Reduction, as described below in Sect. 7.5

7.4 The Mirror-Symmetric Case (Algorithm R)

The generation of orbit cycles requires that the points $t_0u_0v_0w_0$ don't lie in a plane. We can guarantee that such 4-tuples exist, unless the edge figures are perfectly mirror-symmetric: The perpendicular bisector of every edge uv in G acts as a mirror, reflecting the neighbors t of u to the neighbors w of v. Since each edge tu and each edge vw has the same mirror-symmetry, the mirror images of the mirrors are also mirrors. It follows that the component of G that contains u is the orbit of u under the group generated by the mirror reflections for the edges incident to u.

Such groups, groups that are generated by reflections, are called Coxeter groups, and they have been classified in all dimensions, cf. Coxeter (1973), Table IV on p. 297. In four dimensions, there are eight such groups, which are related to the regular polytopes of 4-space, plus an infinite class of *reducible* groups, which are direct products two-dimensional Coxeter groups.

We deal with the Coxeter groups as follows. For each group Γ in the finite list, we determine the smallest distance δ such that the neighbors of a point $u \in \mathbb{S}^3$ at distance δ can generate the group Γ. The smallest value $\delta_{\min}$ of these bound implies, by a packing argument, that $|A|$ is bounded by a constant, and thus we can resort to dimension reduction.

For the infinite family of reducible groups, we are able to identify the two complementary 2-dimensional planes corresponding to the two factor groups, and thus we can replace each component of $G(A)$ by two circles. We process these circles like the the circles that result from orbit cycles (Algorithm M, Sect. 7.3).

7.5 2+2 Dimension Reduction

The classical dimension reduction procedure applies when the image of a *point* in A (or a line through the origin) is known. The image of the complementary 3-dimensional hyperplane is then also known, and we call this *1+3 dimension reduction*. By contrast, in *2+2 dimension reduction*, we have identified a two-dimensional plane P for the point set A and another two-dimensional plane Q for B, and we are looking for congruences that map P to Q, besides mapping A to B.

We first choose a joint coordinate system x_1, y_1, x_2, y_2 in which P and Q coincide with the x_1y_1-plane. The allowable rotations are therefore restricted to independent rotations in the x_1y_1-plane (by some angle φ) and in the complementary x_2y_2-plane (by some angle ψ). After introducing polar coordinates in the two planes, the problem reduces to *translational* congruence between two point sets $\hat{A}$ and $\hat{B}$ on the two-dimensional torus $[0, 2\pi)^2$. The distance components of the polar coordinates are attached as a *label* to each point on the torus, and only points with equal label can be mapped to each other.

We now apply a sequence of condensation and relabeling steps, using Voronoi diagrams on the torus, which eventually lead to *canonical sets* $\hat{A}_0$ and $\hat{B}_0$. These sets play the same role as the canonical axes of Sect. 2 for the problem of a single rotation (or "translation on the one-dimensional torus"): If A and B are congruent (under the constraint of mapping P to Q), then we can choose arbitrary points $a \in \hat{A}_0$ and $b \in \hat{B}_0$, and the unique rotation that maps a to b will map A to B. We therefore have to test only a single candidate rotation.

8 The Four-Dimensional Point Groups

It is tempting to extend the high-level "characterization" of three-dimensional point groups of Theorem 1 to four dimensions:

Conjecture 4. A four-dimensional point group is either

1. the symmetry group of one of the five four-dimensional regular solids,
2. a direct product of lower-dimensional point groups,
3. or a subgroup of one of the above groups.

The four-dimensional point groups have been enumerated, first by Threlfall and Seifert (1931) for the case of direct congruences only (determinant +1), and most lately by Conway and Smith (2003). The book of Conway and Smith gives an explicit list of these groups (Tables 4.1–4.3, pp. 44–47). Thus, in principle, it should be a trivial matter to settle Conjecture 4. However, these groups are specified algebraically, and it is not easy to see geometrically what they are.

When we started our work, we hoped that our techniques would shed light on Conjecture 4, as was the case for three dimensions (Theorem 1), but so far, the implications of our algorithm are not so strong. (On the other hand, the analysis of our algorithm *uses* the classification of four-dimensional finite Coxeter groups, i.e., those point groups that are *generated by reflections*.)

It would also be interesting to see to what extent Conjecture 4 generalizes to higher dimensions. The regular polytopes are known in all dimensions. However, in eight dimensions, the root lattice E_8 has symmetries that don't come from regular polytopes, thus providing counterexamples to a straightforward generalization of Conjecture 4 for eight dimensions, and most likely also for six and seven dimensions.

References

Akutsu, T.: On determining the congruence of point sets in d dimensions. Computational Geometry: Theory and Applications 4(9), 247–256 (1998),

Alt, H., Mehlhorn, K., Wagener, H., Welzl, E.: Congruence, similarity, and symmetries of geometric objects. Discrete & Computational Geometry 3(1), 237–256 (1988), http://dx.doi.org/10.1007/BF02187910

Atkinson, M.D.: An optimal algorithm for geometrical congruence. J. Algorithms 8(2), 159–172 (1987), http://dx.doi.org/10.1016/0196-6774(87)90036-8

Brass, P., Knauer, C.: Testing the congruence of d-dimensional point sets. International Journal of Computational Geometry and Applications 12(1–2), 115–124 (2002), http://dx.doi.org/10.1142/S0218195902000761

Bentley, J.L., Shamos, M.I.: Divide-and-conquer in multidimensional space. In: Proc. Eighth Annual ACM Symp. on Theory of Computing. pp. 220–230. STOC'76, ACM, New York, NY, USA (1976), http://doi.acm.org/10.1145/800113.803652

Conway, J.H., Hardin, R.H., Sloane, N.J.A.: Packing lines, planes, etc.: Packings in Grassmannian spaces. Experiment. Math. 5, 139–159 (1996), https://projecteuclid.org/euclid.em/1047565645

Conway, J.H., Smith, D.A.: On Quaternions and Octonions. A K Peters (2003)

Coxeter, H.S.M.: Regular Polytopes. Dover Publications, 3rd edn. (1973)

Hopf, H.: Über die Abbildungen der dreidimensionalen Sphäre auf die Kugelfläche. Mathematische Annalen 104, 637–665 (1931), http://www.digizeitschriften.de/dms/img/?PID=GDZPPN002274760

Iwanowski, S.: Testing approximate symmetry in the plane is NP-hard. Theoretical Computer Science 80(2), 227–262 (1991), http://dx.doi.org/10.1016/0304-3975(91)90389-J

Kim, H., Rote, G.: Congruence testing of point sets in 4-space, In: Proc. 32st International Symposium on Computational Geometry (SoCG 2016), LIPIcs, to appear

Manacher, G.: An application of pattern matching to a problem in geometrical complexity. Information Processing Letters 5(1), 6–7 (1976), http://dx.doi.org/10.1016/0020-0190(76)90092-2

Sugihara, K.: An $n \log n$ algorithm for determining the congruity of polyhedra. Journal of Computer and System Sciences 29(1), 36–47 (1984), http://dx.doi.org/10.1016/0022-0000(84)90011-4

Threlfall, W., Seifert, H.: Topologische Untersuchung der Diskontinuitätsbereiche endlicher Bewegungsgruppen des dreidimensionalen sphärischen Raumes. Mathematische Annalen 104(1), 1–70 (1931), http://dx.doi.org/10.1007/BF01457920