

CONGRUENCE TESTING FOR POINT SETS IN 4-SPACE

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von

Heuna Kim

2016

Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
heunak@mi.fu-berlin.de

gefördert durch die Deutsche Forschungsgemeinschaft im Rahmen
des Graduiertenkollegs "*Methods for Discrete Structures*" (GRK 1408)

Betreuer: Prof. Dr. Günter Rote
Zweitgutachter: Prof. Dr. Ulrich Brehm
Datum der Disputation: Juni 20. 2016

Abstract

Congruence is the geometric concept of being the same up to rotations and translations in Euclidean space. As congruence is a fundamental concept in geometry, it has drawn broad attentions from the computational geometry community for a long time whether the curse of dimensionality applies to congruence testing. We developed a deterministic optimal-running-time algorithm for congruence testing in 4-space.

To understand the importance of the main algorithm in the historical context, we provide a survey about the computational model and the previous work on congruence testing algorithms. The crucial ingredients of the algorithm are explained component by component. These include general 4-dimensional rotations, angles between linear subspaces, and the Plücker embedding. In the sequence of steps in the algorithm, high regularities are forced in the structure of point sets. This lets us encounter beautiful mathematical structures on a 3-sphere and the symmetry group of finite points: the Hopf fibration of a 3-sphere and the Coxeter group of four-dimensional point groups. We also give an elementary and self-contained overview about these two mathematical topics.

The main algorithm consists of five modules that are interesting in their own right. The algorithm is complicated and we provide rather pessimistic estimates. This algorithm, however, can be regarded as a big step forward to constructing a more efficient algorithm in higher dimensions.

In the same vein, the last part is devoted to the extendability of the algorithm to higher dimensions. This part concludes with discussing implementability and geometric properties that the algorithm may imply.

Acknowledgments

What I confronted, experienced, and learned during my Ph.D study became a part of the most invaluable memories in my life. I would like to thank people, without whom neither such time nor work was possible.

I would like to begin by thanking my advisor, Günter Rote. He let me realize how delightful doing research is and taught me how to do research “right” with his humbleness and honesty. I would like to thank Professor Ulrich Brehm for agreeing to review this thesis. I also would like to thank Otfried Cheong for rescuing me and introducing me a chance to pursue my Ph.D with Günter.

Many thanks should go to my coauthors, Eunjin Oh, Michael Dobbins, Edgardo Roldán-Pensado, Wolfgang Mulzer: I learned a lot from them. Passionate and inspiring discussions with them will not be forgotten. In addition, I wish to thank Luis Montejano, Xavier Goaoc, Heekap Ahn and Christian Knauer for generous supports for research visits.

I was also grateful to work at the AG Theoretische Informatik. I really appreciate countless helpful advice, and sympathy from Rafel Jaume Deyá, Andrei Asinowski, Till Miltzow especially while I was struggling for my Ph.D. I would like to say many thanks to Yannik Stein, Paul Seiferth, Romain Grunert, Bahareh Banyassady for never hesitating whenever I ask help even if it makes them into troubles. They made my Ph.D time smoother and more fun. I would like to thank Matthias Henze, Claudia Dieckmann for being the nicest officemates and Helmut Alt, Frank Hoffmann, Klaus Kriegel for their helpful wisdom.

I would like to gratefully acknowledge the scholarship by the RTG “Methods for Discrete Structures”.

I was a happy graduate student thanks to Kate Shin, Jihye Hong, Nevena Palić, Nieke Aerts, Eva Martínez, who gave me cheerful energy to finish my thesis during lunch time and via skype. Above all, I would like to thank my parents, my brother and David, who always stood behind me without any questions, for everything.

March 2016, Berlin.
Heuna

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction to Congruence Testing | 1 |
| 1.1 | Problem Statement | 1 |
| 1.2 | The Computational Model | 1 |
| 1.3 | Previous Algorithms | 2 |
| 1.3.1 | Congruence Testing in the Plane | 2 |
| 1.3.2 | In Three-dimensional Space | 3 |
| 1.3.3 | Congruence Testing in Four- or Higher-dimensional Space | 4 |
| 1.3.4 | More Related Work | 6 |
| 1.4 | Contributions | 6 |
| | I Extending Geometric Tools | 9 |
| 2 | Reilluminating Four-dimensional Geometry | 13 |
| 2.1 | Four-dimensional Rotations | 13 |
| 2.2 | An Angle between a Pair of Planes | 14 |
| 2.3 | Plücker Space and Plücker Distance | 17 |
| 2.4 | The Construction of Hopf Fibrations | 20 |
| 2.4.1 | Construction by Isoclinic Rotations. | 20 |
| 2.4.2 | Equivalence of an Invariant Family and a Hopf Bundle. | 21 |
| 2.4.3 | Properties of the Hopf fibration | 22 |
| 2.5 | The Coxeter Classification of Four-dimensional Point Groups | 24 |
| 2.5.1 | The Radius of an Inscribed Sphere of a Fundamental Region | 25 |
| 3 | Auxiliary Tools | 29 |
| 3.1 | Useful Principles | 29 |
| 3.1.1 | The Pruning and Condensing Principles | 29 |
| 3.1.2 | Packing Arguments | 30 |
| 3.2 | Representation Details | 31 |
| 3.2.1 | Canonical Axes | 31 |
| 3.2.2 | Representation of Congruence Types | 32 |
| 3.3 | Leaping From Previous Ideas | 33 |
| 3.3.1 | 1 + 3 dimension Reduction | 33 |
| 3.3.2 | Closest-pair Graphs | 33 |
| 3.3.3 | Finding Representative Points from a 2-sphere: Algorithm K | 34 |

| | | |
|------------|---|-----------|
| II | The New Algorithm in 4-space | 35 |
| 4 | Overview of the New Algorithm | 39 |
| 5 | Five Crucial Modules | 43 |
| 5.1 | Iterative Condensing Based on the Closest-Pair Graph: Algorithm C | 43 |
| 5.2 | Generating Orbit-Cycles: Algorithm O | 46 |
| 5.3 | The Mirror Case: Algorithm R | 48 |
| 5.4 | Marking and Condensing of Great Circles: Algorithm M | 50 |
| 5.5 | 2+2 Dimension Reduction: Algorithm T | 53 |
| 5.5.1 | A Canonical Set Procedure | 53 |
| III | More Implications | 57 |
| 6 | Further Directions to Higher-dimensional Space | 61 |
| 6.1 | In 5 Dimensions | 61 |
| 6.2 | In High Dimensions | 62 |
| 7 | Concluding Remarks | 65 |
| 7.1 | Implementability | 65 |
| 7.2 | Regularity and Related Open Questions | 65 |
| 7.3 | Summary and Future Work | 66 |
| | Bibliography | 69 |

Introduction to Congruence Testing

As well as being intuitive, congruence is a very fundamental notion in mathematics. The well-known congruence testing methods for triangles go back to 300 BC, written in Euclid's Elements. Although this method was known for dozens of centuries, algorithms for testing congruence for point sets have been started to develop only several decades ago [28, 4, 20, 34, 5, 2, 10, 1]. Furthermore, it has been believed that there should be an efficient algorithm for testing congruence in any fixed dimension. Nevertheless, there have been no such an algorithm for four or higher dimensions since lots of two-dimensional and three-dimensional algorithms were devised around 1970's–1980's [28, 4, 20, 34, 5, 2].

In the following sections, more precise definitions of the problem and the computational model are described, and previous work is summarized. This chapter wraps up with stating contributions of this thesis and organization of the remaining chapters.

1.1 Problem Statement

We are given two n -point sets $A, B \subset \mathbb{R}^d$. Our aim is to test whether they are *congruent*, i. e., whether there exist an orthogonal matrix and a translation vector t such that $B = RA + t$. We may only consider orthonormal matrices R of determinant $+1$ (“direct congruences”, rotations), because if we want to allow orthonormal matrices of determinant -1 (“mirror congruence”, rotations and a reflection) as well, we can just repeat the algorithm with a reflected copy of B . We also may consider only whether $B = RA$ by translating A and B so that their centroids are moved to the origin.

1.2 The Computational Model

Because the output of problem is sensitive to numerical errors, it is also natural to consider an approximate version of the congruence testing problem. However, since congruence testing with error tolerances is known to be NP-hard [16, 24], we restrict our concern to the exact case. (The algorithm in this thesis is also expected to run for congruence testing with an error tolerance in a special setting. This case is discussed in Chapter 7.)

In order to test exact congruence, we assume that we can perform exact arithmetic with real numbers. In order to meet this assumption, which is common in Computational Geometry, we use the *Real Random-Access Machine (Real-RAM) model* (for example in [31, Chapter 1.4]). In this model, we can freely use square roots, sines and cosines, and basic operations from linear algebra such as eigenvalues of 2×2 -matrices (or any matrices of constant size), and we assume that the computational model gives exact results in constant time. Sines and cosines and their inverses can be eliminated in favor of purely algebraic operations.

If we had restricted our model to rational arithmetic, this would severely constrain the

problem. For example, in the plane, the only rotational symmetries that a point set with rational coordinates can have are multiples of 90° . A three-fold symmetry is possible in 3 dimensions, but a fivefold symmetry is impossible in any dimension. An input with limited symmetry, however, would not be interesting enough to consider. The previous algorithms for the exact congruence testing problem [28, 4, 20, 34, 5, 2] were developed around 70's–80's under the same assumption. Those algorithms, however, did not mention the assumption explicitly except in [5, 2]. All these algorithms are surveyed in the next section.

1.3 Previous Algorithms

It has been shown that the complexity of the problem is $\Omega(n \log n)$ independently by Atallah (1985) [4], Highnam (1986) [20], and Atkinson (1987) [5]. This bound holds even in one-dimensional space. It can be easily seen by reducing the problem to the set equality problem in the real line.

It has been believed that $O(n \log n)$ algorithms should exist for any fixed dimension; in other words, the congruence testing problem should be fixed-parameter tractable with the dimension parameter. Nevertheless, the previous best deterministic algorithm by Brass and Knauer (2000) [10] for d -space for $d > 3$ took time $O(n^{\lceil d/3 \rceil} \log n)$. The previous best randomized algorithm by Akutsu (1998) [1] takes time $O(n^{\lfloor d/2 \rfloor / 2} \log n)$ for $d \geq 6$ and time $O(n^{3/2} \log n)$ for $d = 4, 5$. Therefore, the previous best result in 4-space was time $O(n^2 \log n)$ deterministically and time $O(n\sqrt{n} \log n)$ with randomization.

In this section, ideas of optimal algorithms in two- and three-dimensional space are first summarized in Section 1.3.1 and in Section 1.3.2. Lastly, algorithms that work for any dimensions but whose complexities are exponential to the dimension are surveyed.

1.3.1 Congruence Testing in the Plane

Manacher (1976) [28]. The first algorithm is by Manacher. After translating point sets so that the centroid lies on the origin O , his algorithm sorts all input points by spherical coordinates (r, θ) , first by angle θ , and secondly by distance r to the origin O in both increasing order. Let $\{p_i = (r_i, \theta_i) | i = 1, \dots, n\}$ be the given point set indexed by the sorted order. Finally, the algorithm generates a cyclic sequence in which r_i and $\angle p_i O p_{i+1} = \theta_{i+1} - \theta_i$ alternate for $1 \leq i \leq n$: $\langle r_1, \theta_2 - \theta_1, r_2, \dots, \theta_n - \theta_{n-1}, r_n \rangle$. Point sets are congruent if and only if the corresponding cyclic sequences are the same up to cyclic shifts. His paper [28] mainly discusses geometric applications of the algorithm by Knuth, Morris and Pratt [27] that determines whether a string y contains a string x as a substring in linear time $O(|x| + |y|)$. It can be determined if two sequences are the same up to cyclic shifts by duplicating one sequence and checking if the original sequence is a subsequence of the duplicated sequence by using the algorithm by Knuth, Morris and Pratt in linear time. Manacher's algorithm takes time $O(n \log n)$ due to sorting.

Atallah (1985) [4]. The main goal of Atallah's algorithm is to enumerate all the axes of mirror symmetries in the plane. Once we know all the axes of mirror symmetries of two point sets, any axis in one point set should be matched to any axis in another point set by some congruence mapping. Afterward, the rotation angle of congruence mapping can be figured out and congruence can be also tested in linear time. Note that any axis of mirror symmetries goes through the centroids O . The algorithm classifies points by the distance to the centroid O , computes the axes for each class and obtains the intersection of the axes for all the classes; this generates the axes of symmetries of a given point set. For each class with the same distance to the centroid O , the algorithm generates a cyclic sequence as follows. Let $\{p_i\}$ be points in such a class sorted by spherical

coordinates in increasing order. Then, the cyclic sequence repeats $\angle(p_iOp_{(i \bmod n+1)})$ twice for $i = 1, \dots, n$: $\langle \angle(p_1Op_2), \angle(p_1Op_2), \dots, \angle(p_{n-1}Op_n), \angle(p_{n-1}Op_n), \angle(p_nOp_1), \angle(p_nOp_1) \rangle$. Any cyclic shift of this sequence that becomes a palindrome corresponds to an axis of mirror symmetry. Then, it enumerates all the palindromic cyclic shifts of the sequence by using the algorithm by Knuth, Morris and Pratt [27] in linear time. The time complexity of Atallah’s algorithm is $O(n \log n)$ as well because it classifies and sorts points according to the distance from the centroid.

Highnam (1986) [20]. Highnam’s algorithm is mainly to find axes of mirror symmetries and rotational symmetries. To find mirror symmetry, his algorithm looks up palindromes but does not classify input by the distance to the centroid. Instead, it uses the sequence similar to Manacher’s algorithm. It also uses the algorithm by Knuth, Morris and Pratt to find palindromic locations. By the similar reasoning, the total time complexity is $O(n \log n)$.

1.3.2 In Three-dimensional Space

Sugihara(1984) [34]. Sugihara’s algorithm determines whether two bounded polyhedra in 3-space, i.e., closed subsets in 3-space with finite volume bounded by a finite number of flat polygons, are congruent in time $O(E \log E) = O(n \log n)$ where E is the number of edges in the polyhedra. Once whether two polyhedra in 3-space are congruent can be determined in time $O(n \log n)$, congruence for two point sets in 3-space can be determined in $O(n \log n)$; we apply Sugihara’s algorithm for the convex hull of each class of point sets of the same distance from the origin, and apply again after radially projecting all input points to the same sphere and taking convex hulls. Two point sets are congruent if and only if every pair of classes and projections are all congruent.

Sugihara’s algorithm extends Hopcroft and Tarjan’s algorithm [21] to test graph isomorphism by regarding a polyhedron as an embedding of a graph. Note that a convex polytope, i.e., a bounded convex polyhedron, can be regarded as an embedding of a three-connected planar graph. Hopcroft and Tarjan’s algorithm is only for three-connected planar graphs but Sugihara’s is applicable for more general graphs. Therefore, Sugihara’s algorithm is applicable for bounded polyhedra.

Sugihara’s algorithm partitions directed copies of input edges into “mutually indistinguishable” edges in $O(n \log n)$ time: basically, it means that one edge can be mapped to another by congruence mapping. More precisely, in Sugihara’s paper, two edges e, e' are mutually indistinguishable if any traversing t of directed edges following only the rightmost- or the leftmost-adjacent edge and another traversing that has the same structure to the traversing t can be related by relating e and e' . The main idea of this partitioning is by the following steps.

- (1) Calculate a quadruple for each directed edge e , consisting of the length of e , the angle defined by two adjacent faces of e , the angle defined by e and the rightmost-adjacent edge to e , and the angle defined by e and the leftmost-adjacent edge to e .
- (2) Classify edges by values of their quadruples.
- (3) Subdivide classes of edges more by traversing directed paths of edges.
- (4) Compare resulting classes.

Hopcroft and Tarjan’s algorithm [21] makes sure that the subdivision procedure is done until each class becomes mutually indistinguishable and comparing mutually indistinguishable classes is enough for testing congruence.

Atkinson(1987) [5]. Atkinson’s algorithm is to determine congruence for two point sets in 3-space. This algorithm begins with reducing input point sets as small as possible while preserving

symmetries and then considers a bounded number of all the possible matches of the original point sets, obtained by the reduced sets. The principle of this reduction procedure is different from *pruning* mentioned before (refer to Section 1.2 and Section 3.1.1) but is more related to a *canonical set procedure* (refer to Section 3.2.1 and Section 5.5.1). The first part of this algorithm can be considered a canonical set procedure of for a 2-sphere with rotational symmetries. This reduction procedure preserves symmetries, whereas pruning does not. For reduction, Atkinson's algorithm constructs a closest-pair graph, i. e., a graph whose vertices are points and edges are pairs of vertices that achieve the minimum distance (see Section 3.3.2). Then, the algorithm prunes points by the congruence type of a neighborhood of a vertex v , or by a *vertex figure* (refer to Section 3.2.2). Then, either there is a component of the centroid different from the original centroid or the degrees of vertices are bounded since v has at most five closest vertices by the kissing number (see Section 3.1.2). The sets can be further reduced by traversing each component if the degree is two and by comparing faces if the degree is three or four or five. At the end, there are only two cases:

- (i) the reduced set is a singleton set or consists of two antipodal points.
- (ii) the reduced set has a bounded cardinality more than two.

For (ii), the algorithm finds all the rotations from a pair of non-antipodal points in one reduced set to any pair of non-antipodal points at the same angle in another reduced set. Then, it checks if at least one of such rotations transforms one input point set to another input point set. For (i), the rotational axis can be identified so Manacher's algorithm [28] (see the first paragraph of Section 1.3.1) can be applied after each point is represented by a cylindrical coordinate regarding the common rotational axis as the z -axis.

The variation of the first part of Atkinson's algorithm will be used in the new algorithm and discussed again in Lemma 3.2 in Section 3.3.3.

Alt, Mehlhorn, Wagener, and Welzl(1988) [2]. The algorithm by Alt, Mehlhorn, Wagener, and Welzl checks graph isomorphism as well. On top of this, their algorithm uses labeling and that a convex-hull of points in 2-sphere can be regarded as a three-connected planar graph. First, the algorithm projects all the points radially to the unit sphere centered at the centroid and labels them with distances from the centroid. Then, it computes the convex hull of the projected points. Also, it labels each projected point p with the lexicographically-smallest cyclic shift of the string which enumerates the distances and angles of neighboring points of p in the convex hull. If two labeled graphs are isomorphic, then two original input point sets are congruent. Labeled graph isomorphism for the three-connected planar graph is determined by applying the partitioning algorithm of Hopcroft and Tarjan [21], [36, Section 4.13] in time $O(n \log n)$. The total time complexity is $O(n \log n)$ as well.

1.3.3 Congruence Testing in Four- or Higher-dimensional Space

Alt, Mehlhorn, Wagener, and Welzl (1988) [2]. This algorithm reduces a d -dimensional congruence testing problem to n subproblems in $(d-1)$ -space. As in the three-dimensional case of their algorithm, it projects all the points radially to the $(d-1)$ -dimensional unit sphere S centered at the centroid and labels them with distances to the centroid. Fix one point a in the resulting set from A . The next step is another projection to a $(d-2)$ -dimensional unit sphere S' which is the intersection of S and the hyperplane that orthogonally bisects the line segment from a to the centroid; project each point x in A except a onto S' along the arc from x to a on the surface of S . Let us denote the resulting set as A' . We can obtain sets B'_1, \dots, B'_n in the same manner by fixing all points of B . Two given point sets A and B are congruent if and only if A' is congruent to at least one of B'_i for $i = 1, \dots, n$ with taking labels into account. As a

result, we obtained n subproblems to determine congruence between A' and B'_i for $i = 1, \dots, n$. Thus, the algorithm by Alt, Mehlhorn, Wagener, and Welzl takes time $O(n^{d-2} \log n)$ in d -space.

Akutsu(1998) [1]: a Randomized Version. Akutsu developed $O(n^{(d-1)/2} \log n)$ -time randomized algorithm for congruence testing problem in d -space and he stated that his algorithm achieves the time complexity $O(n^{d/4+O(1)})$ by acknowledging Matoušek's idea. Matoušek's unpublished idea is to match closest pairs, i. e., the pairs of points that attain the minimum distance, (refer to Section 3.3.2) instead of input points, since a closest pair together with the centroid spans a 2-plane and the orthogonal space of a 2-plane is $(d-2)$ -dimensional if the ambient space is d -dimensional. This idea improves the algorithm by Alt, Mehlhorn, Wagener, and Welzl by time $O(n^{\lfloor \frac{d}{2} \rfloor} \log n)$ as well.

Akutsu's algorithm also reduces the original problem to multiple lower-dimensional subproblem. With randomization, however, his algorithm chooses $O(\sqrt{n})$ points randomly and projects original input points to a hyperplane H by fixing one point among randomly chosen points as an orthogonal vector to H . At the end, this generates two collections of projected sets in 3-space such that (a) if there is a congruent pair from each collection, the two given input sets need to be congruent and (b) if not, the probability that they are congruent is very low due to the birthday paradox. Instead of comparing every pair from each collection, by using the canonical form and lexicographical sorting, whether there exists a pair of two congruent sets can be determined efficiently in time $O(n^{(d-1)/2} \log n)$ without Matoušek's idea. The accurate analysis with Matoušek's idea was not provided but still the algorithm has the bounded error probability after modification since the analysis for the error probability does not depend on the dimension reduction but uses only the birthday paradox. Then, Akutsu's algorithm obtains the time complexity $O(n^{\lfloor d/2 \rfloor} \log n)$ for $d \geq 6$ and $O(n^{3/2} \log n)$ for $d = 4, 5$ with Matoušek's idea.

The projection step of Akutsu's algorithm is also different from that of Alt, Mehlhorn, Wagener and Welzl. Instead of projecting points radially, Akutsu's algorithm projects all points to the hyperplane perpendicular to the segment \overline{cp} from the centroid to p where p is one of the points chosen randomly before.

During the projection step, labeling is done simultaneously in a way that two points have the same label if there is an orientation-preserving isometry between these two points that fixes the segment \overline{cp} . A variant of this projection method is used in the new algorithm and discussed in Section 3.3.1.

Brass and Knauer(2003) [10]. Brass and Knauer's algorithm extends Matoušek's idea by fixing a triple of points that contains two closest pairs instead of fixing a closest pair. A triple of points together with the centroid can determine a 3-dimensional subspace and its orthogonal space is $(d-3)$ -dimensional unless the triple and the centroid are coplanar. By projecting points to a $(d-3)$ -dimensional orthogonal subspace, the algorithm can achieve time $O(n^{\lfloor d/3 \rfloor} \log n)$, but the actual algorithm requires more consideration due to inner symmetries of triples.

There are examples of point sets $A \subset \mathbb{R}^4$ where a triple of points can be mapped to other $\Omega(n^2)$ triples in A by isometries (see Figure 1.1). This tricky case, that is, when fixing a triple can have too many alternative reductions, can happen only when the point sets lie in orthogonal 2-planes and all the triples are coplanar. To deal with this case, the algorithm constructs connected components. Each component starts with an antipodal pair of points and components of the

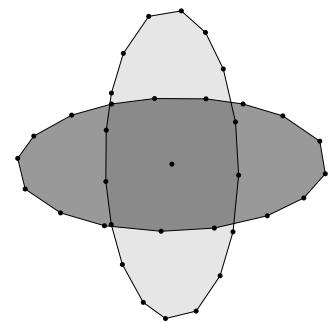


Figure 1.1: If all the closest-pairs are in orthogonal 2-planes, any triple can be mapped to other $\Omega(n^2)$ triples by rotations.

smallest distance merge if their distance is smaller than $\sqrt{2}$. This merging process repeats until either a triple of non-coplanar points in a component is found, or the smallest distance between components turns out to be $\sqrt{2}$. If such a triple is found, the dimension reduction can be applied. Otherwise, the given point set is in orthogonal 2-planes. This is because the smallest distance between two antipodal pairs in the unit sphere can reach the maximum ($\sqrt{2}$) if and only if they are orthogonal pairs. In this case, for each component pair, they apply congruence testing algorithm for the plane and check if there exists a matching of components such that each matched pair is congruent.

1.3.4 More Related Work

Jiang, Yu, and Bunke (1991) [25] provided an algorithm that can test symmetries of general 3-dimensional polyhedral objects in time $O(n^2 \log n)$. Also, there is another paper by Brass and Knauer (2004) [11] for testing congruence for general 3-dimensional objects in time $O(n \log n)$. Their algorithm can be applied for the case that Sugihara's algorithm [34] cannot be applied. Rezende and Lee (1995) [15] developed an algorithm that tests a k -subset congruence in $O(kn^d)$ with considering scaling in d -dimension together.

1.4 Contributions

Congruence is a fundamental geometric problem, and it has a long history as seen in Section 1.3. It is believed by many researchers in the field that the problem is solvable in $O(n \log n)$ time in any *fixed* dimension. However, so far the best algorithms in general dimension d are exponential in d . The new algorithm is the first 4-dimensional optimal algorithm.

The previously discussed algorithms are rather simple compared to the new algorithm. The new algorithm uses many geometric tools intensively to exploit structure in 4-space. In the course of employing such tools, not only are these tools extended but also new geometric insights are provided and interesting characteristics are revealed. All those tools are intertwined with the new algorithm. We elaborate all the necessary backgrounds to understand how these tools function as ingredients of the new algorithm in Part I.

As a result of symmetries, we will also encounter beautiful mathematical structures. In particular, the structure of Hopf bundles organizes the special case of isoclinic planes in a pleasant way. One of the main importances in this part is to describe the interpretation of the Hopf fibration in a very geometric and elementary way by using four-dimensional rotations. This interpretation clarifies the necessary and sufficient condition for a collection of great circles to be in a Hopf bundle.

This part includes preliminaries about four-dimensional rotations, angles between a pair of planes, Plücker coordinates, four-dimensional Coxeter groups, kissing numbers and representing a neighborhood of a vertex or an edge in 4-space.

The new algorithm finally comes on the stage in Part II. The techniques of closest pairs, pruning, and dimension reduction have been used for congruence testing before. We extend these ideas and apply them in a novel way. Our algorithm uses closest pairs not just as a convenient tool for matching, but as a device for scrutinizing an actual structure of a closest-pair graph (i. e., a graph whose vertices are points and edges indicate the minimum distance). Our algorithm extracts helices around great circles on 3-sphere by taking advantage of the structure of the closest-pair graph.

The subalgorithm in Section 5.4 of pruning these extracted great circles is interesting in its own right; this subalgorithm includes the step that, given a set of great circles, selects a subset of great circles in the same Hopf bundle.

On top of the previous dimension reduction principle, a new efficient dimension reduction technique (called 2+2 dimension reduction, see Section 5.5) is developed in the new algorithm for the case that a given point set has rotational symmetries that make two orthogonal planes invariant. This new dimension reduction technique is simple to implement and resolves the trickiest case in Brass and Knauer's algorithm as well [10]. We would expect that 2+2 dimension reduction can be extended to higher dimensions (see Section 6.2) and applied to other problems that require dimension reduction.

I

Extending Geometric Tools

This part plays a role not only in introducing the necessary background for understanding the new algorithm, but also in providing alternative perspectives to four-dimensional geometry and extracting new applicable ideas.

This part consists of two chapters: one about 4-dimensional geometry and another about more tools. Even though most of these tools are an extension of previous work, many ideas are new.

Because this part is a collection of ingredients for the new algorithm, the sections in this part may seem independent. Because they have complicated interdependences, it is not easy to see these only by text. Still it is useful and important to see the big picture about which modules in the new algorithm are supported by these sections. The diagram below (Figure 1.2) summarizes the relations between sections in Part I and those in Part II.

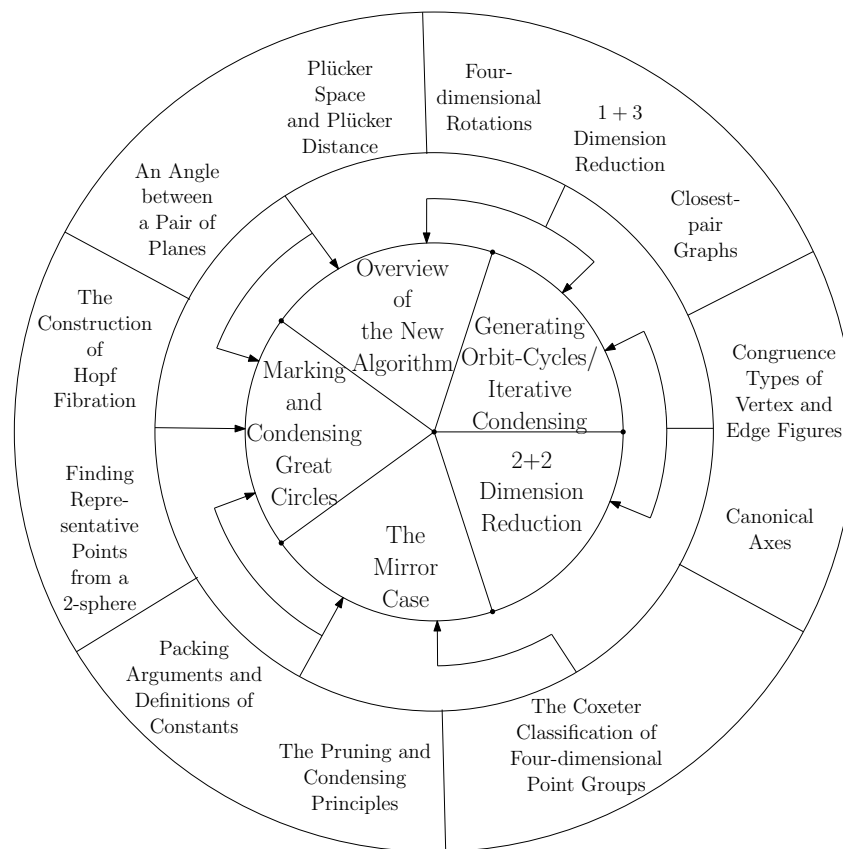


Figure 1.2: Interdependence between Part I and Part II. Part I is placed in the outer ring and Part II is in the inner disk.

Reilluminating Four-dimensional Geometry

2.1 Four-dimensional Rotations

A congruence mapping, or orientation-preserving isometry, is a combination of rotations and translations. As mentioned in Section 1.1, the influence of translations is no longer a concern for us, but rotations are still tricky to handle; this section provides geometric intuitions of general four-dimensional rotations while using only the simple tool of representing them.

We can classify rotations in 4-space into two categories: simple rotations and double rotations. A simple rotation is a two-dimensional rotation in 4-space in which one entire plane through the origin is fixed. A double rotation is a rotation that makes a pair of two orthogonal planes P, Q invariant (i. e., P is mapped to P itself and Q is mapped to Q itself by this rotation) and all the half-lines from the origin in P (and Q) are displaced by the angles φ (and ψ) inside P (and Q) for $\varphi, \psi \in (-\pi, \pi)$. Note that simple rotations are special cases of double rotations by letting $\varphi = 0$ or $\psi = 0$.

These geometric intuitions of 4-dimensional rotations can be encoded by a 4×4 rotation matrix R . A 4×4 rotation matrix R is an orthogonal matrix with determinant $+1$. It has four eigenvalues of absolute value 1, whose product is $+1$. They come in conjugate complex pairs, $e^{\pm i\varphi}$ and $e^{\pm i\psi}$. The special case of real eigenvalues 1 or -1 is included. Here, φ and ψ in the eigenvalues correspond to φ and ψ in the angular displacements.

Let us assume first that the four eigenvalues are distinct. This case is called a non-isoclinic rotation. The eigenvectors for each conjugate complex pair, which are conjugate complex, span a real 2-plane. We choose an orthonormal basis v_1, v_2 for the plane corresponding to $e^{\pm i\varphi}$, and an orthonormal basis v_3, v_4 for the plane corresponding to $e^{\pm i\psi}$. In the resulting orthonormal basis v_1, v_2, v_3, v_4 , the matrix R looks as follows:

$$R = R_{\varphi, \psi} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & \cos \psi & -\sin \psi \\ 0 & 0 & \sin \psi & \cos \psi \end{pmatrix} \quad (2.1)$$

The pair $\{P, Q\}$ of orthogonal planes $P = \langle v_1, v_2 \rangle$ and $Q = \langle v_3, v_4 \rangle$ is uniquely determined by R , but the basis v_1, v_2 and v_3, v_4 for each plane is not unique. Here, P and Q are the only pair of planes that are invariant under the rotation. We can require that the basis v_1, v_2, v_3, v_4 is positively oriented by flipping v_3 and v_4 and changing from $R_{\varphi, \psi}$ to $R_{\varphi, -\psi}$ if necessary (or by flipping v_1 and v_2 and changing to $R_{-\varphi, \psi}$, if we prefer). The angle pair $\{\varphi, \psi\}$ in the range $-\pi < \varphi, \psi < \pi$ is then unique up to a simultaneous negation to $\{-\varphi, -\psi\}$.

If the four eigenvalues are not distinct, they come in equal pairs $e^{i\varphi}, e^{i\varphi}, e^{-i\varphi}, e^{-i\varphi}$. Such rotations are called *isoclinic rotations*. We exclude the special cases $1, 1, 1, 1$ and $-1, -1, -1, -1$

(the identity and the inversion) and assume $0 < \varphi < \pi$. We can still find a basis v_1, v_2, v_3, v_4 for which the matrix has the form $R = R_{\varphi, \varphi}$ but the decomposition into two orthogonal planes is not unique. There are infinitely many pairs of orthogonal planes that are invariant under the rotation, but still not all the pairs of planes in 4-space are invariant. If we insist that the basis is positively oriented, the rotation can either be written as $R_{\varphi, \varphi}$ or $R_{\varphi, -\varphi}$. A rotation of the first class is called a *right-isoclinic rotation*, and the second class a *left-isoclinic rotation*.

Axis Planes. For a rotation R that is not isoclinic, there is a pair of orthogonal planes $P = \langle v_1, v_2 \rangle, Q = \langle v_3, v_4 \rangle$ that is invariant under the rotation R . We call these two planes P, Q axis planes of the rotation R . Given a rotation matrix R , we can find axis planes P and Q by computing v_1, v_2, v_3, v_4 as follows. Let $u_\varphi, u_{-\varphi}, u_\psi, u_{-\psi}$ be eigenvectors of R such that u_θ corresponds to the eigenvalue $e^{i\theta}$ where $\theta \in \{\varphi, -\varphi, \psi, -\psi\}$. Then, $u_\varphi = v_1 + iv_2, u_{-\varphi} = v_1 - iv_2$ and $u_\psi = v_3 + iv_4, u_{-\psi} = v_3 - iv_4$ are conjugate pairs. Then,

$$v_1 = \frac{u_\varphi + u_{-\varphi}}{2}, \quad v_2 = \frac{u_\varphi - u_{-\varphi}}{2i},$$

$$v_3 = \frac{u_\psi + u_{-\psi}}{2}, \quad v_4 = \frac{u_\psi - u_{-\psi}}{2i}.$$

2.2 An Angle between a Pair of Planes

In Section 2.1, we have seen that a pair of two orthogonal planes can be used to represent a non-isoclinic rotation. Not only for representing rotations but also for understanding four-dimensional geometry in general, a pair of 2-planes is worthwhile to investigate. In particular, a linear subspace of Euclidean 4-space \mathbb{R}^4 is either a line or a 3-space (a hyperplane) or a 2-plane. A line and a hyperplane is relatively easier to be taken care of: A pair of lines is determined up to congruence by the smallest angle between them and so is a pair of hyperplanes by the smallest angle between their normal vectors.

In this section, we restrict our concerns to only planes that go through the origin. Our objective is, given a pair of 2-planes in 4-space, to find a feature invariant under rotations and translations.

Geometric Definition Using Half Rays. We first define an angle between two planes in 4-space geometrically and then show the computation method. The angle between two planes P, Q is defined as a pair α, β of two real numbers in range $[0, \pi/2]$ where α and β are defined as follows [29].

We define α as the minimum angle between any two half lines Op in P and Oq in Q . Such an α exists [29]. We denote such half lines that make the minimum angle α as Op^* and Oq^* . Let U be the plane that contains Op^* and Oq^* . The plane U is perpendicular to the planes P and Q , since otherwise Op^* and the projection of Op^* to Q would make an angle smaller than α . Let U' be the orthogonal plane to U . Then U' is also perpendicular to P and Q . Let ℓ_p be the intersection of P and U' . Similarly, let ℓ_q be the intersection of Q and U' . We define the acute angle made by the lines ℓ_p and ℓ_q as β . See Figure 2.1.

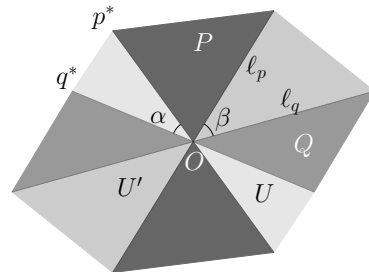


Figure 2.1: Angles between a pair of planes in 4-space.

Isoclinic Planes. We just mentioned that there exists the minimum angle α between any pairs of half lines, each of which is in the plane P and the plane Q respectively. However, the pair of half lines Op^* and Oq^* that defines the angle α may not be unique but an infinite number of such pairs can exist. If there are an infinite number of such pairs, there exist an infinite number of perpendicular planes U to P and Q . In this case, an infinite number of planes U cut out P and Q by making all the equal angles [29]. Thus, $\alpha = \beta$. We say that a pair of planes of angle α, α where $0 \leq \alpha \leq \pi/2$ is isoclinic.

Computation. This can be formulated as follows. Let L_P and L_Q be the vector spaces that span P and Q respectively. Then,

$$\begin{aligned}\cos \alpha &= \max_{\substack{u_1 \in L_P, \\ |u_1|=1}} \max_{\substack{v_1 \in L_Q, \\ |v_1|=1}} u_1^T v_1 = (u_1^*)^T (v_1^*), \\ \cos \beta &= \max_{\substack{u_2 \in L_P, \\ |u_2|=1, \\ u_2^T u_1^*=0}} \max_{\substack{v_2 \in L_Q, \\ |v_2|=1, \\ v_2^T v_1^*=0}} u_2^T v_2.\end{aligned}$$

Let M_P and M_Q be the orthonormal basis of P and Q . The previous equation boils down to

$$\max_{\substack{u \in L_P, \\ |u|=1}} \max_{\substack{v \in L_Q, \\ |v|=1}} u^T v = \max_{\substack{y \in \mathbb{R}^2, \\ |y|=1}} \max_{\substack{z \in \mathbb{R}^2, \\ |z|=1}} y^T (M_P^T M_Q) z.$$

By the minimax characterization of the singular value decomposition (SVD), the angle can be computed by the SVD as follows:

$$\begin{pmatrix} \cos \alpha & 0 \\ 0 & \cos \beta \end{pmatrix} = Y^T (M_P^T M_Q) Z.$$

This computation can be found [9].

With this definition, an angle between two planes in 4-space is invariant under rotations and translations.

Equivalent Definition by Orthogonal Projections. Let P, Q be a pair of planes and let C' be an orthogonal projection of a unit circle in P to Q . The maximum and the minimum distances $\cos \alpha, \cos \beta$ from a point on C' to the origin define the same angle α and β . According to this definition, generally the orthogonal projection C' is an ellipse with the following reasoning. If P and Q are not isoclinic, there is a pair of planes $\{U, U'\}$ that are perpendicular to both P and Q and cut out P and Q at the angle α and β respectively. Then, the intersection of Q and U , (or the line containing the half line Oq^*) is the major axis (longer axis) of the ellipse because U defines the closest distance between any half lines in P and Q . Similarly, the intersection of Q and U' , (or ℓ_q) is the minor axis (shorter axis) of the ellipse. However, for a pair of isoclinic planes, the orthogonal projection C' becomes a circle. The argument is shown in Proposition 2.1.

The following procedure will be useful in Section 5.4. Let us assume P and Q are not isoclinic and in addition to C' , let D' be an orthogonal projection of a unit circle in Q to P . Then, C' and D' are ellipses. A pair of points in P can be *marked* by the intersection of the unit circle on P and the major axis of D' . Similarly, a pair of points in Q can be marked as well.

Left and Right Pairs for Isoclinic Planes. Let $P = \langle v_1, v_2 \rangle, Q = \langle u_1, u_2 \rangle$ be a pair of 2-planes in 4-space where v_1, v_2 are orthonormal vectors that span P and u_1, u_2 are unit vectors parallel to the projections of v_1, v_2 to Q . Extend v_1, v_2 to a positively-oriented basis

(v_1, v_2, v_3, v_4) in 4-space; that is, a quadruple of orthonormal vectors such that the determinant of $(v_1, v_2, v_3, v_4) = +1$. Consider the projections u'_1 and u'_2 of u_1 and u_2 in Q to the v_3v_4 -plane. If u'_1 and u'_2 are positively oriented together with v_1, v_2 , we say a pair $\{P, Q\}$ is a right pair, otherwise a left pair. In other words, if the projection of u_1, u_2 is oriented in the same way as the v_3, v_4 -plane is oriented, $\{P, Q\}$ is a right pair and if they are oriented in the opposite way, $\{P, Q\}$ is a left pair. Whether P, Q is a right or left pair does not depend on the order of P and Q . Determining if a pair of two planes are a right pair or a left pair can be done by checking the sign of the determinant of $(v_3v_4)^T(u_1u_2)$. If it is positive, it is a right pair, and otherwise, it is a left pair. This classification as right or left is called the *chirality* of a pair of isoclinic planes. When $\alpha = \beta = \pi/2$ (completely orthogonal planes) or $\alpha = \beta = 0$ (identical planes), the pair of planes is both a left- and right-pair.

Relations to Clifford Parallelism. A pair (C, D) of great circles in a 3-sphere \mathbb{S}^3 is called Clifford parallel if for any point p in C , the shortest distance from p to D is the same. A pair (C, D) of great circles is Clifford parallel if and only if a pair (P, Q) of planes spanned by C and D is isoclinic. This relation is stated as the following proposition.

Proposition 2.1. *For a given pair of planes P and Q , the following three statements are equivalent.*

1. P and Q are isoclinic.
2. Unit circles C on P and D on Q are Clifford parallel.
3. The projection C' of a unit circle C on P to Q is a circle and so is the projection D' of a unit circle D on Q to P .

Proof. [1 \Leftrightarrow 2.] The planes P and Q are at angle α, α if and only if there are an infinite number of pairs of half lines, each pair of which consists of a half line Op on P and a half line Oq on Q that form an angle α . Also, these pairs of lines $\{Op, Oq\}$ define the pairs of points in C and D with the shortest distance. For any pair $\{Op, Oq\}$, the intersection p' of Op and C , the intersection q' of Oq and D , and the origin form congruence triangles. Then, the distance between p' and q' is constant for any points p' on C and q' on D . Thus, the planes P and Q are at angle α, α if and only if unit circles C, D on P and Q are Clifford parallel.

[1 \Leftrightarrow 3.] In the previous paragraph, it is explained why the projections C' and D' generally form ellipses. Similarly to the previous argument, points in Op are orthogonally projected to points in Oq for the previously defined pairs $\{Op, Oq\}$ since any plane perpendicular to both P and Q cuts out P and Q with the same angle α . Therefore, the projections of any points in C to Q are equidistant from the origin and so are the projections of any points in D to P . Hence, P and Q are isoclinic if and only if the projection C' and D' are circles. \square

Terminology for a pair of planes carries over to a pair of great circles, such as an angle of great circles, a right pair of great circles, and a left pair of great circles. For example a pair of great circles in a 3-sphere \mathbb{S}^3 is at angle (α, β) if the pair of planes spanned by them is at angle (α, β) for $0 \leq \alpha, \beta \leq \pi/2$ as well.

A right/left pair of great circles are also known as Clifford parallel of the first/second kind [8, 7]. For convenience, we simply say that a pair of great circles is a right/left pair instead of Clifford parallel of the first/second kind.

The new algorithm generates a set of great circles on a 3-sphere \mathbb{S}^3 and relates each great circle C to the plane that C spans. Later, it will turn out if a pair of great circles are not Clifford parallel, we can do the marking procedure, illustrated in the previous paragraph “Equivalent Definition by Orthogonal Projections”. Otherwise, this pair defines a unique Hopf fibration.

Generating Clifford-parallel Circles. The following useful lemma shows how to parametrize Clifford-parallel circles. This lemma is used to prove important properties in Section 2.4.3.

Lemma 2.2. *Let v_1, v_2, v_3, v_4 be a positively-oriented orthonormal basis and let C be the great circle in a 3-sphere \mathbb{S}^3 spanned by two orthonormal vectors v_1 and v_2 . If C, D forms a right pair of great circles in \mathbb{S}^3 at angle (α, α) , then D is spanned by the two orthonormal vectors*

$$\begin{aligned} u_1(\delta) &= v_1 \cos \alpha + (v_3 \cos \delta + v_4 \sin \delta) \sin \alpha \\ u_2(\delta) &= v_2 \cos \alpha + (v_4 \cos \delta - v_3 \sin \delta) \sin \alpha \end{aligned} \quad (2.2)$$

for some $\delta \in [0, 2\pi)$, and it can be parameterized as

$$\begin{aligned} \cos \gamma \cdot u_1(\delta) + \sin \gamma \cdot u_2(\delta) &= \\ v_1 \cos \alpha \cdot \cos \gamma + v_2 \cos \alpha \cdot \sin \gamma + v_3 \sin \alpha \cdot \cos(\delta + \gamma) + v_4 \sin \alpha \cdot \sin(\delta + \gamma) \end{aligned}$$

with parameter $0 \leq \gamma < 2\pi$. Similarly, for a left pair, D is spanned by

$$\begin{aligned} u'_1(\delta) &= v_1 \cos \alpha + (v_3 \cos \delta + v_4 \sin \delta) \sin \alpha \\ u'_2(\delta) &= v_2 \cos \alpha + (v_3 \sin \delta - v_4 \cos \delta) \sin \alpha \end{aligned}$$

for some $\delta \in [0, 2\pi)$.

Proof. The circle D can be written as $\{ \cos \gamma \cdot u_1 + \sin \gamma \cdot u_2 \mid 0 \leq \gamma < 2\pi \}$, for any choice of two orthonormal vectors $u_1, u_2 \in D$. In terms of the basis v_1, v_2, v_3, v_4 , we may write $u_1 = \sum_{i=1}^4 a_i v_i$ and $u_2 = \sum_{i=1}^4 b_i v_i$ with

$$\sum_{i=1}^4 a_i^2 = \sum_{i=1}^4 b_i^2 = 1, \quad \sum_{i=1}^4 a_i b_i = 0. \quad (2.3)$$

The projection of D on the $v_1 v_2$ -plane containing C is then the curve $\cos \gamma \cdot (a_1 v_1 + a_2 v_2) + \sin \gamma \cdot (b_1 v_1 + b_2 v_2)$. By the definition isoclinic planes of angle α, α , this projection must be a circle of radius $\cos \alpha$. Thus, the two vectors $a_1 v_1 + a_2 v_2$ and $b_1 v_1 + b_2 v_2$ must be perpendicular vectors of length $\cos \alpha$. We may choose u_1 in such a way that its projection $a_1 v_1 + a_2 v_2$ becomes the vector $v_1 \cos \alpha$. This implies $a_1 = \cos \alpha, a_2 = 0, b_1 = 0, b_2 = \pm \cos \alpha$. By flipping u_2 if necessary, we can achieve that $b_2 = +\cos \alpha$.

We have $a_1^2 + a_2^2 = b_1^2 + b_2^2 = \cos^2 \alpha$ and $a_1 b_1 + a_2 b_2 = 0$. In view of (2.3), this implies $a_3^2 + a_4^2 = b_3^2 + b_4^2 = \sin^2 \alpha$ and $a_3 b_3 + a_4 b_4 = 0$. Thus, the two vectors (a_3, a_4) and (b_3, b_4) are orthogonal vectors of norm $\sin \alpha$. The general form of such vectors is $(a_3, a_4) = \sin \alpha \cdot (\cos \delta, \sin \delta)$ and $(b_3, b_4) = \pm \sin \alpha \cdot (-\sin \delta, \cos \delta)$. It can be checked that the choice of the positive sign leads to a right pair, and the other choice leads to a left pair. \square

2.3 Plücker Space and Plücker Distance

The importance of dealing with 2-planes in 4-space for the new algorithm is already emphasized enough from previous sections. One classical and effective tool of dealing with linear subspaces in a vector space is the Grassmannian and the Plücker embedding. The *Grassmannian* $\mathbb{G}(g, V)$ is the collection of all g -dimensional linear subspaces of a vector space V . The *Plücker embedding* embeds Grassmannians into higher-dimensional projective space. In this section, we restrict our consideration only to the Grassmannian $\mathbb{G}(2, \mathbb{R}^4)$ and Plücker coordinates, that is, six-dimensional homogeneous coordinates of $\mathbb{G}(2, \mathbb{R}^4)$ in real projective 5-space.

The Grassmannian $\mathbb{G}(2, \mathbb{R}^4)$ is the set of all the planes going through the origin in Euclidean 4-space and equivalent to lines in real projective 3-space. Let $x = (x_0, x_1, x_2, x_3), y =$

(y_0, y_1, y_2, y_3) be homogeneous coordinates of two distinct points in real projective 3-space. Then a Plücker coordinate for a line going through these two points is the sextuple $(p_{ij})_{0 \leq i < j \leq 3}$ where $p_{ij} = x_i y_j - x_j y_i$; in other words, the sextuple of the 2×2 determinants of the 2×4 matrix with rows x and y . The Plücker coordinate is also regarded as a homogeneous coordinate in projective 5-space since it is uniquely determined by a line in projective 3-space up to a nonzero scaling. For convenience, we call real projective 5-space with Plücker coordinates *Plücker space*. Each plane in Euclidean 4-space corresponds to a point in Plücker space.

To use a metric for 2-planes in Euclidean 4-space, we define a *Plücker distance* in Plücker space using Plücker coordinates. Plücker distance between two planes P, Q in 4-space is the squared Euclidean distance between normalized Plücker coordinates of P and Q , taking the sign of scaling factors into account. Plücker distance is defined in a way such that it satisfies the distance axiom (the conditions for a metric), since it is a basically Euclidean distance.

Because Plücker space is real projective 5-space \mathbb{P}^5 and \mathbb{P}^5 can be geometrically regarded as a 5-sphere with having antipodal points identified, or $\mathbb{P}^5 \simeq \mathbb{S}^5 / \mathbb{Z}^2$, the framework of the Plücker distance can be conceived as embedding planes into antipodal pairs of points in a 5-sphere \mathbb{S}^5 and measuring the smallest Euclidean distance between two antipodal pairs in \mathbb{S}^5 .

The next lemma shows that the Plücker distance is geometrically meaningful and does not depend on the choice of a coordinate system. This is shown by relating angles between planes to the Plücker distance.

Lemma 2.3. *If a pair of planes P, Q is at angle (α, β) , with $0 \leq \alpha, \beta \leq \pi/2$, the Plücker distance between them is $\sqrt{2}(1 - \cos \alpha \cos \beta)^{1/2}$.*

Proof. Let P and Q be two planes with angle (α, β) . Let $u = (u_1, u_2, u_3, u_4), v = (v_1, v_2, v_3, v_4)$ be orthonormal vectors that span P . If P and Q are not orthogonal, let $u' = (u'_1, u'_2, u'_3, u'_4), v' = (v'_1, v'_2, v'_3, v'_4)$ be the orthonormal vectors such that u', v' are parallel to the projections of u, v to Q . If P and Q are orthogonal, that is, $\alpha = \beta = \pi/2$, let u', v' be any two orthonormal vectors that span Q .

The Plücker coordinates are then the six values $(k_{ij})_{0 \leq i < j \leq 3}$, where $k_{ij} = u_i v_j - v_i u_j$. Since u, v is an orthonormal basis, we can check that these Plücker coordinates are already normalized, using the relations $k_{ij} = -k_{ji}$ and $k_{ii} = 0$:

$$\begin{aligned} \sum_{0 \leq i < j \leq 3} k_{ij}^2 &= \frac{1}{2} \sum_{0 \leq i < j \leq 3} (k_{ij}^2 + k_{ji}^2) = \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 k_{ij}^2 \\ &= \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 (u_i v_j - v_i u_j)^2 = \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 (u_i^2 v_j^2 + v_i^2 u_j^2 - 2u_i u_j v_i v_j) \\ &= \frac{1}{2} \left(\sum_{i=0}^3 u_i^2 \sum_{j=0}^3 v_j^2 - 2 \sum_{i=0}^3 u_i v_i \sum_{j=0}^3 u_j v_j + \sum_{j=0}^3 u_j^2 \sum_{i=0}^3 v_i^2 \right) \\ &= \frac{1}{2} (1 - 2 \cdot 0 + 1) = 1 \end{aligned}$$

Similarly, the Plücker coordinates $\ell_{ij} = u'_i v'_j - v'_i u'_j$ of Q are normalized: $\sum_{0 \leq i < j \leq 3} \ell_{ij}^2 = 1$. The squared Euclidean distance between these points is

$$\sum_{0 \leq i < j \leq 3} (k_{ij} - \ell_{ij})^2 = \sum_{0 \leq i < j \leq 3} k_{ij}^2 + \sum_{0 \leq i < j \leq 3} \ell_{ij}^2 - 2 \sum_{0 \leq i < j \leq 3} k_{ij} \ell_{ij} = 2 \left(1 - \sum_{0 \leq i < j \leq 3} k_{ij} \ell_{ij} \right)$$

Let us evaluate the last term. From the angle between P and Q , we get $u^T u' = \cos \alpha, v^T v' =$

$\cos \beta$, and $u^T v' = v^T u' = 0$, and we have $k_{ij} = -k_{ji}$ and $\ell_{ij} = -\ell_{ji}$.

$$\begin{aligned}
\sum_{0 \leq i < j \leq 3} k_{ij} \ell_{ij} &= \frac{1}{2} \sum_{0 \leq i < j \leq 3} (k_{ij} \ell_{ij} + k_{ji} \ell_{ji}) = \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 k_{ij} \ell_{ij} \\
&= \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 (u_i v_j - v_i u_j)(u'_i v'_j - v'_i u'_j) \\
&= \frac{1}{2} \sum_{i=0}^3 \sum_{j=0}^3 (u_i u'_i v_j v'_j + v_i v'_i u_j u'_j - u_i v'_i u'_j v_j - v_i u'_i u_j v'_j) \\
&= \frac{1}{2} \left(\sum_{i=0}^3 u_i u'_i \sum_{j=0}^3 v_j v'_j + \sum_{i=0}^3 v_i v'_i \sum_{j=0}^3 u_j u'_j \right. \\
&\quad \left. - \sum_{i=0}^3 u_i v'_i \sum_{j=0}^3 u'_j v_j - \sum_{i=0}^3 u'_i v_i \sum_{j=0}^3 u_j v'_j \right) \\
&= \frac{1}{2} ((u^T u')(v^T v') + (v^T v')(u^T u') - (u^T v')(u'^T v) - (u'^T v)(u^T v')) \\
&= \frac{1}{2} (\cos \alpha \cos \beta + \cos \beta \cos \alpha - 0 - 0) = \cos \alpha \cos \beta
\end{aligned}$$

Thus, the Euclidean distance is $\sqrt{2(1 - \cos \alpha \cos \beta)}$. Each circle is represented by two antipodal points. Thus we should consider also negated vectors $(-k_{ij})_{0 \leq i < j \leq 3}$ and $(-\ell_{ij})_{0 \leq i < j \leq 3}$. Negation can be achieved by negating one of the vectors u, v or one of the vectors u', v' . In the formulas, this changes $\cos \alpha$ to $\cos(\pi - \alpha) = -\cos \alpha$ or $\cos \beta$ to $\cos(\pi - \beta) = -\cos \beta$. The minimum distance is obviously achieved when choosing the positive cosine values, i.e., when choosing the angles in the range $0 \leq \alpha, \beta \leq \pi/2$. It is easy to check that this smaller distance value together with the larger value, $\sqrt{2(1 + \cos \alpha \cos \beta)}$, form the sides of a rectangle inscribed in the unit circle, as expected from two pairs of antipodal points on a sphere. \square

Corollary 2.4. *The Plücker distance is invariant under rotations and reflections.*

Proof. The angle between two planes is invariant over rotations and reflections and by Lemma 2.3, since the Plücker distances depend only on the angles between pairs of planes. \square

Comparison with Other Distances. Conway, Hardin and Sloan [12] considered different distances in Grassmannian space. One of the distances that they considered is called the *chordal distance*. For the Grassmannian $\mathbb{G}(2, \mathbb{R}^4)$, the chordal distance is defined as $\sqrt{\sin^2 \alpha + \sin^2 \beta}$ for the pair of planes at angle α, β . The chordal distance coincides with Plücker distance if a pair of planes is isoclinic.

They also considered the *geodesic distance* [37], and the *Asimov's distance* [3, 19]. All those three distances are by embedding the Grassmannian $\mathbb{G}(k, \mathbb{R}^d)$ to a m -sphere where $m = \binom{d+1}{2} - 1$. On the other hand, if the Plücker distance is extended to the Grassmannian $\mathbb{G}(k, \mathbb{R}^d)$, the Plücker distance is measured in the ambient space of dimension $\binom{d}{k}$.

Those three distances are defined by angles in Grassmannian space. This angle is just the extension of an angle between planes in 4-space. By regarding an angle as a vector with k values, the geodesic distance, the chordal distance, and the Asimov's distance are defined as the Euclidean norm of an angle, the Euclidean norm of the sine value of an angle, and the maximum norm of an angle respectively.

Optimal packing is numerically computed for these three distances [12].

2.4 The Construction of Hopf Fibrations

The characteristics of the Hopf fibrations are already well-known. Refer to [8, 7, Section 4.3.7, Chapter 18.8]. Although the Hopf fibrations can be defined for $(2i + 1)$ -spheres and $(4i + 3)$ -spheres for all $i \geq 0$ except a 15-sphere [7, Chapter 18.1.3.6], for the new algorithm we only consider the Hopf fibration [22, 23] of a 3-sphere. The ultimate goal of this section is to understand the relations among isoclinic rotations (in Section 2.1), a left/right pair of great circles (in Section 2.2) and Hopf fibrations.

Before getting started, the flow of this section is sketched without the definitions of terms:

1. We first construct *the invariant family* for a given great circle C by using only isoclinic rotations (Lemma 2.5 and Lemma 2.6).
2. Then, we define a *Hopf map* h with respect to C . This shows that the invariant family for C is equivalent to the *Hopf bundle* for h (Lemma 2.7). This gives a geometric perspective of constructing a *Hopf fibration*.
3. In Section 2.4.3, we derive the following properties.
 - (a) From 1, we know there is a unique bundle for each great circle C (Corollary 2.8). Also, a pair of Clifford-parallel great circles belong to a common Hopf bundle (Lemma 2.9).
 - (b) The relation between two great circles defined by a right pair (or a left pair) is transitive; thus, it is an equivalence relation. From 3a, an equivalence class is actually a right Hopf bundle (or a left Hopf bundle) (Corollary 2.10). The converse is also true.
 - (c) The Hopf fibration maps two great circles at angle α, α to two points at geodesic distance 2α in a 2-sphere (Lemma 2.11).
 - (d) From 3c, a Hopf fibration is unique up to rotations and reflections. This implies independence of choices of a basis (Corollary 2.12).

To minimize confusion, we provide proofs and explanations only for a right Hopf fibration but parallel lemmas are also valid for a left Hopf fibration.

2.4.1 Construction by Isoclinic Rotations.

This construction of a right Hopf fibration provides the relation between a right Hopf fibration and a right-isoclinic rotation. We will now construct a right Hopf bundle as the set of orbits of all right rotations that map a given circle to itself.

We mention that the convention of assigning the label left or right to versions of a structure that come in pairs is somewhat arbitrary and not uniform in the literature. In fact, right Hopf bundles are also connected to *left-isoclinic* rotations: a right Hopf bundle is the set of images of a given circle under all left-isoclinic rotations. In the quaternion representation, right rotations according to the convention in this paper are carried out by left multiplication with unit quaternions. Thus, there are good reasons also for alternative choices.

Lemma 2.5. *For every great circle C in a 3-sphere and two points $p, q \in C$, there exists a unique right-isoclinic rotation R which rotates C in itself and rotates p to q .*

Proof. Choose an orthonormal basis v_1, v_2 of the plane P spanned by C and arbitrarily label one of the orientations of C as positive and the other as negative. There exists a unique rotation

$$\mathbf{R}_\gamma = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix}$$

in the $v_1 v_2$ -plane that rotates p to q . Choose again a basis v_3, v_4 of the plane P^\perp orthogonal to the plane P so that a basis v_1, v_2, v_3, v_4 forms a positively-oriented basis. Here v_3, v_4 are unique up to rotations in P^\perp and the orientation of v_3, v_4 is fixed. Then

the 4×4 matrix

$$\rho(\gamma) = R_{\gamma,\gamma} \quad (2.4)$$

with a basis v_1, v_2, v_3, v_4 gives a unique right-isoclinic rotation that rotates C to itself where $R_{\varphi,\psi}$ is defined in (2.1). \square

The family of right-isoclinic rotations $\mathcal{F} = \{\rho(\gamma) | \gamma \in [0, 2\pi)\}$ rotates a great circle C to itself where $\rho(\gamma)$ is defined as in (2.4). Note that \mathcal{F} is a one-dimensional group isomorphic to the special orthogonal group $\text{SO}(2)$. We call \mathcal{F} the *family of right rotations* for C .

Lemma 2.6. *Let \mathcal{F} be the family of right rotations for some great circle in a 3-sphere \mathbb{S}^3 . For every point p in \mathbb{S}^3 , the orbit of p under rotations in \mathcal{F} is a great circle of \mathbb{S}^3 .*

Proof. If $p = (x, y, z, w)$, the orbit of p generated by a right-isoclinic rotation is

$$\mu(\gamma) = \rho(\gamma)p = u_1 \cos \gamma + u_2 \sin \gamma \quad (2.5)$$

where

$$u_1 = (x, y, z, w)^T, u_2 = (-y, x, -w, z)^T, \rho(\gamma) \in \mathcal{F}$$

for $\gamma \in [0, 2\pi)$. Here, u_1 and u_2 are orthogonal and $\mu(\gamma)$ is a parametrization of a great circle, so p generates a great circle as an orbit. \square

From Lemma 2.6, \mathcal{F} partitions \mathbb{S}^3 into the set of great circles that are invariant under rotations in \mathcal{F} . Let Γ be the set of these great circles and call it the *invariant family* for C . Eventually, the invariant family Γ forms a right *Hopf bundle*. In the remainder of this section, we illustrate what this statement means more precisely.

2.4.2 Equivalence of an Invariant Family and a Hopf Bundle.

A *fibration* is a map that projects a *fiber bundle* to a *base*, by identifying a subspace of the fiber bundle, called a *fiber*, to a point in a base in a way that any point in the fiber bundle is contained in exactly one fiber.

Given a great circle C , by choosing a Cartesian coordinate system (x, y, z, w) such that a great circle C lies in the x, y -plane, we define a right Hopf map with respect to C as

$$h(x, y, z, w) = (2(xw - yz), 2(yw + xz), 1 - 2(z^2 + w^2)) \quad \text{for} \quad (x, y, z, w) \in \mathbb{S}^3 \subset \mathbb{R}^4$$

Note that such a Cartesian coordinate system is not unique but we choose it arbitrarily. For reference, a left Hopf map can be defined as $h'(x, y, z, w) = (2(xw + yz), 2(yw - xz), 1 - 2(z^2 + w^2))$.

The following lemma shows that a right Hopf map is a fibration.

Lemma 2.7. *Let C be a great circle in a 3-sphere \mathbb{S}^3 . Let Γ be the invariant family for C . Let h be the right Hopf map with respect to C . Then, for any $D \in \Gamma$, h maps all points in D to one point in the 2-sphere \mathbb{S}^2 .*

Proof. The map h indeed maps the 3-sphere to the 2-sphere, since $(1 - 2z^2 - 2w^2)^2 + 4(xw - yz)^2 + 4(yw + xz)^2 = (x^2 + y^2 + z^2 + w^2)^2 = 1$.

As we have seen in Lemma 2.6, for any $D \in \Gamma$, if $p = (x, y, z, w) \in D$, we can parametrize D as $\mu(\gamma)$ in (2.5) for $\gamma \in [0, 2\pi)$. Then,

$$\begin{aligned}
h(\mu(\gamma)) &= h((x, y, z, w) \cos \gamma + (-y, x, -w, z) \sin \gamma) \\
&= h((x \cos \gamma - y \sin \gamma, y \cos \gamma + x \sin \gamma, z \cos \gamma - w \sin \gamma, w \cos \gamma + z \sin \gamma)) \\
&= (2[(x \cos \gamma - y \sin \gamma)(w \cos \gamma + z \sin \gamma) - (y \cos \gamma + x \sin \gamma)(z \cos \gamma - w \sin \gamma)], \\
&\quad 2[(y \cos \gamma + x \sin \gamma)(w \cos \gamma + z \sin \gamma) + (x \cos \gamma - y \sin \gamma)(z \cos \gamma - w \sin \gamma)], \\
&\quad 1 - 2[(z \cos \gamma - w \sin \gamma)^2 + (w \cos \gamma + z \sin \gamma)^2]) \\
&= (2[(xw \cos^2 \gamma - yz \sin^2 \gamma + (xz - yw) \cos \gamma \sin \gamma) \\
&\quad - (yz \cos^2 \gamma - xw \sin^2 \gamma + (xz - yw) \cos \gamma \sin \gamma)], \\
&\quad 2[(yw \cos^2 \gamma + xz \sin^2 \gamma + (xw + yz) \cos \gamma \sin \gamma) \\
&\quad + (xz \cos^2 \gamma + yw \sin^2 \gamma - (xw + yz) \cos \gamma \sin \gamma)], \\
&\quad 1 - 2[(z^2 \cos^2 \gamma + w^2 \sin^2 \gamma - zw \cos \gamma \sin \gamma) \\
&\quad + (w^2 \cos^2 \gamma + z^2 \sin^2 \gamma + zw \cos \gamma \sin \gamma)]) \\
&= (2(xw - yz), 2(yw + xz), 1 - 2(z^2 + w^2))
\end{aligned}$$

This is independent of γ , which shows that D is a fiber.

This also implies that a right Hopf map is a fibration, since elements in Γ , or fibers, are mutually disjoint and cover the whole \mathbb{S}^3 . Remember that in Lemma 2.6 every point in \mathbb{S}^3 is in a unique orbit, so in a unique element of Γ , under rotations in \mathcal{F} . \square

Indeed, a right Hopf map with respect to a great circle C is a fibration that maps an invariant family Γ for C to a 2-sphere \mathbb{S}^2 in a way that each great circle in Γ is mapped to a point in \mathbb{S}^2 . Hereafter, we are allowed to call a Hopf map a Hopf fibration, the fiber bundle Γ the Hopf bundle, and each great circle in Γ a Hopf fiber. Also, the 2-sphere which is a base for a Hopf fibration is called a base sphere. We say a right Hopf fibration/bundle/fiber if a Hopf fibration/bundle/fiber was induced by a right Hopf map.

2.4.3 Properties of the Hopf fibration

Because the invariant family for a great circle C is equivalent to a right Hopf bundle, we may as well state Lemma 2.5 as in the following corollary.

Corollary 2.8 (Corollary of Lemma 2.5). *Every great circle belongs to a unique right Hopf bundle.*

Next, we show the relation between great circles and a Hopf fibration.

Lemma 2.9. *Any right pair of great circles belongs to a common right Hopf bundle.*

Proof. Let C, D be a right pair of great circles at angle α, α . Let h be a right Hopf map for C . With the same basis for h , C is represented by $\{(\cos \gamma, \sin \gamma, 0, 0) | \gamma \in [0, 2\pi)\}$. Then h maps C to the north pole since $h(\cos \gamma, \sin \gamma, 0, 0) = (0, 0, 1)$. The circle D can be represented by $\{(\cos \alpha \cdot \cos \gamma, \cos \alpha \cdot \sin \gamma, \sin \alpha \cdot \cos(\delta + \gamma), \sin \alpha \cdot \sin(\delta + \gamma)) | \gamma \in [0, 2\pi)\}$ for some $\delta \in [0, 2\pi)$ by

Lemma 2.2. Thus, h maps D to a point $(\sin 2\alpha \sin \delta, \sin 2\alpha \cos \delta, \cos 2\alpha)$, since

$$\begin{aligned}
& h(\cos \alpha \cos \gamma, \cos \alpha \sin \gamma, \sin \alpha \cos(\delta + \gamma), \sin \alpha \sin(\delta + \gamma)) \\
&= \left(2(\cos \alpha \sin \alpha \cos \gamma \sin(\delta + \gamma) - \cos \alpha \sin \alpha \sin \gamma \cos(\delta + \gamma)), \right. \\
&\quad \left. 2(\cos \alpha \sin \alpha \sin \gamma \sin(\delta + \gamma) + \cos \alpha \sin \alpha \cos \gamma \cos(\delta + \gamma)), \right. \\
&\quad \left. 1 - 2(\sin^2 \alpha \cos^2(\gamma + \gamma_0) + \sin^2 \alpha \sin^2(\delta + \gamma)) \right) \\
&= (\sin 2\alpha \sin \delta, \sin 2\alpha \cos \delta, \cos 2\alpha). \tag{2.6}
\end{aligned}$$

This is independent of the parameter γ . Therefore, the right Hopf bundle given by h contains both C and D as Hopf fibers. \square

This relation of being a right pair is actually transitive.

Corollary 2.10. *Let us denote $C \parallel_+ D$, when two great circles C and D in \mathbb{S}^3 form a right pair. Then \parallel_+ is transitive. Each equivalence class of \parallel_+ is a right Hopf bundle.*

Proof. If $C \parallel_+ D$ and $D \parallel_+ E$, there is a unique right Hopf fibration containing C , D and E by Corollary 2.8 and Lemma 2.9. Then, C , D , and E are contained in the same Hopf fibration, so $C \parallel_+ E$. Observe that \parallel_+ is also reflexive and symmetric. The equivalence class is exactly the invariant family for C , or a right Hopf bundle. \square

In our construction, the Hopf map h depends on the choice of a great circle C and a Cartesian coordinate system (x, y, z, w) . However, the Hopf fibration resulting from the map is independent of this choice. To show this, we show how a pair of Hopf fibers is mapped to a pair of points in a base sphere. A similar statement can be found in [7, Exercise 18.11.18].

Lemma 2.11. *Two right Hopf fibers at angle α, α are mapped to a pair of points of geodesic distance 2α by the right Hopf map.*

Proof. By (2.6) in Lemma 2.9, the great circle that is mapped to $\begin{pmatrix} \sin 2\gamma \sin \delta \\ \sin 2\gamma \cos \delta \\ \cos 2\gamma \end{pmatrix}$ by the right

Hopf map is spanned by the two orthonormal vectors

$$u(\gamma, \delta) = \begin{pmatrix} \cos \gamma \\ 0 \\ \cos \delta \sin \gamma \\ \sin \delta \sin \gamma \end{pmatrix}, \quad v(\gamma, \delta) = \begin{pmatrix} 0 \\ \cos \gamma \\ -\sin \delta \sin \gamma \\ \cos \delta \sin \gamma \end{pmatrix}.$$

Let us calculate the angle α between two great circles C and D with parameters (γ_1, δ_1) and (γ_2, δ_2) . By rotation we may assume that $\delta_1 = 0$.

The length of the projection of any point of C onto the plane spanned by D is $\cos \alpha$. We project $u(\gamma_1, 0)$ on the plane with orthonormal basis $u(\gamma_2, \delta_2)$ and $v(\gamma_2, \delta_2)$, and get

$$\begin{aligned}
\cos^2 \alpha &= \langle u(\gamma_1, 0), u(\gamma_2, \delta_2) \rangle^2 + \langle u(\gamma_1, 0), v(\gamma_2, \delta_2) \rangle^2 \\
&= (\cos \gamma_1 \cos \gamma_2 + \cos \delta_2 \sin \gamma_1 \sin \gamma_2)^2 + (-\sin \delta_2 \sin \gamma_1 \sin \gamma_2)^2 \\
&= \cos^2 \gamma_1 \cos^2 \gamma_2 + \sin^2 \gamma_1 \sin^2 \gamma_2 + 2 \cos \delta_2 \cos \gamma_1 \cos \gamma_2 \sin \gamma_1 \sin \gamma_2 \\
&= \cos^2 \gamma_1 \cos^2 \gamma_2 + (1 - \cos^2 \gamma_1)(1 - \cos^2 \gamma_2) + \frac{1}{2} \cos \delta_2 \sin 2\gamma_1 \sin 2\gamma_2 \\
&= 2 \cos^2 \gamma_1 \cos^2 \gamma_2 + 1 - \cos^2 \gamma_1 - \cos^2 \gamma_2 + \frac{1}{2} \cos \delta_2 \sin 2\gamma_1 \sin 2\gamma_2 \\
&= \frac{1}{2}(2 \cos^2 \gamma_1 - 1)(2 \cos^2 \gamma_2 - 1) + \frac{1}{2} + \frac{1}{2} \cos \delta_2 \sin 2\gamma_1 \sin 2\gamma_2 \\
&= \frac{1}{2}(\cos 2\gamma_1 \cos 2\gamma_2 + 1 + \cos \delta_2 \sin 2\gamma_1 \sin 2\gamma_2) \tag{2.7}
\end{aligned}$$

We get

$$\cos 2\alpha = 2 \cos^2 \alpha - 1 = \cos 2\gamma_1 \cos 2\gamma_2 + \sin 2\gamma_1 \sin 2\gamma_2 \cos \delta_2 \quad (2.8)$$

from (2.7).

On the other hand, we compute the spherical distance c between the two points in the base sphere corresponding to $(\gamma_1, 0)$ and (γ_2, δ_2) . It is the third side of a spherical triangle with angle $\delta_2 - 0 = \delta_2$ at the north pole and sides $a = 2\gamma_1$ and $b = 2\gamma_2$. By the spherical cosine law,

$$\begin{aligned} \cos c &= \cos a \cos b + \sin a \sin b \cos \delta_2 \\ &= \cos 2\gamma_1 \cos 2\gamma_2 + \sin 2\gamma_1 \sin 2\gamma_2 \cos \delta_2 \end{aligned}$$

From this and (2.8), we conclude that $c = 2\alpha$.

Therefore, the angular distance between C and D after applying a Hopf map is 2α . \square

Corollary 2.12. *The images of a set of great circles \mathcal{S} in a 3-sphere under a right Hopf map are independent of choices of a great circle or a Cartesian coordinate system. The right Hopf bundle is unique up to rotations and reflections.*

Proof. A map from one image of \mathcal{S} to another image is an isometry since it preserves geodesic distances between pairs of points in the image of \mathcal{S} . \square

The following theorem summarizes the relations of right pairs and isoclinic rotations. The analogous statement is valid for left pairs as well.

Theorem 2.13. *For a pair of great circles C and D , C and D is a right pair at angle α , α if and only if D is an orbit under all right rotations that leave C invariant.*

Proof. By Lemma 2.5 and Lemma 2.6, there exists the family \mathcal{F} of right rotations that leave C invariant and orbits of \mathcal{F} are great circles of a 3-sphere. By Lemma 2.7, the set of these orbits is equivalent to the set of fibers in the (unique) right Hopf bundle containing C . Thus, a circle D is an orbit of \mathcal{F} if and only if C and D are in the common right Hopf bundle. Lemma 2.9 implies that C and D are indeed in the common right Hopf bundle. \square

2.5 The Coxeter Classification of Four-dimensional Point Groups

The new algorithm treats the case that a given point set can be generated by a series of reflections as a special case. Our algorithm uses the classification of Coxeter groups to argue that the cardinalities of point sets that have a high degree of symmetry are bounded. To this end, this section provides the classification of such groups in 4-space and shows the computation of inradii of fundamental regions of such groups.

The discrete groups generated by mirror reflections have been classified in arbitrary dimension, cf. [14, Table IV on p. 297]. Each of these groups of rank d can be represented by a collection of mirrors r_1, \dots, r_d such that $r_i \cdot r_i = 1$ and $(r_i r_j)^{m_{ij}} = 1$ for $i \neq j$ and $m_{ij} \geq 2$; such a group is called a Coxeter group. The value of m_{ij} specifies the dihedral angle between mirrors r_i and r_j as π/m_{ij} . The Coxeter diagram, or the Coxeter graph, explicitly encodes these values of m_{ij} .

In four dimensions, Coxeter groups were first enumerated by Édouard Goursat in 1899: There are five *irreducible groups*, which are the symmetry groups of the six regular 4-dimensional polytopes and the demihypercube, called A_4, C_4, B_4, F_4 and G_4 according to [14] (or alternatively A_4, BC_4, D_4, F_4 , and H_4 in today's terms). Here, A_m denotes a group related to reflectional symmetries of a m -simplex. Similarly, C_m is for that of a m -hypercube and a m -cross-polytope,

B_m is for that of a m -demihypercube. F_4 represents the reflectional symmetry group for a 24-cell and G_4 is for an 120-cell and a 600-cell.

In addition, there are the *reducible* groups, direct products of lower-dimensional reflection groups. They are the groups $A_3 \times A_1$, $C_3 \times A_1$, $G_3 \times A_1$, and $D_2^p \times D_2^q$, for $p, q \geq 2$, where G_3 is the symmetry group of the icosahedron and the dodecahedron, C_3 is the symmetry group of the cube and the octahedron, and D_2^p is the dihedral group of order $2p$, (denoted as $I_2(p)$ alternatively,) the symmetry group of the regular p -gon. Some groups have alternative representations: $D_3 = C_3$, $D_2^2 = A_1 \times A_1$, $D_2^3 = A_2$, and $D_2^4 = C_2$. For reducible groups, their mirrors fall into two or more classes such that each mirror in one class is perpendicular to the mirrors in the other classes. Since reflections at perpendicular mirrors commute, this is how reducible groups yield the decomposition into a direct product of smaller groups.

The arrangement of all mirror hyperplanes of a reflection group cuts the 3-sphere into equal cells, which can be taken as the *fundamental regions* of the group. These fundamental regions are not necessarily equal to the Voronoi regions of the point set; the Voronoi regions are usually cut into smaller cells by mirrors passing through the centers of the regions. It is known that the fundamental region of a reflection group is a spherical *simplex*, see [14, Theorem 11.23]. Thus, in 4-space, the fundamental region is a spherical tetrahedron T , and the group is generated by four independent reflections.

The inradii of fundamental regions of Coxeter groups are of our particular interest for Section 5.3.

2.5.1 The Radius of an Inscribed Sphere of a Fundamental Region

By using the Coxeter diagrams [14, Table IV on p. 297] and Table 2.1, we know the dihedral angles between bounding mirrors of fundamental regions of Coxeter groups. Each vertex in a Coxeter diagram represents mirrors. Any pair of vertices with no edge means a pair of perpendicular mirrors. An edge with no number connects a pair of mirrors whose dihedral angle is $\pi/3$. Edges with number p implies a dihedral angle π/p .

Among all four-dimensional Coxeter groups, we are interested in the finite list, excluding the infinite family $D_2^p \times D_2^q$; that is, $A_4, C_4, B_4, F_4, G_4, A_3 \times A_1, C_3 \times A_1$, and $G_3 \times A_1$. Given the dihedral angles, normal vectors of the vertices of a spherical tetrahedron T can be computed, since when the dihedral angle between two mirrors i and j is π/m_{ij} ,

$$\hat{n}_i \cdot \hat{n}_j = \cos(\pi - \pi/m_{ij}) = -\cos(\pi/m_{ij})$$

where \hat{n}_i, \hat{n}_j are unit normal vectors to mirrors i and j and \cdot is an inner product.

Table 2.1 enumerates such normal vectors for the above groups. The unit normal vector of the i th bounding mirror in a diagram is in the i th row. For example, for A_4 , the first two bounding mirrors form an angle $\pi/3$ and the corresponding unit normal vectors in the outward directions from the fundamental region form an angle $\pi - \pi/3$. That is why the inner product of the first two normal vectors for A_4 is $\cos(2\pi/3) = -\cos(\pi/3)$.

We call the distance from the center of the inscribed sphere of a spherical tetrahedron T to the mirrors the *Euclidean radius* of T . We compute the Euclidean radius R_0 of each spherical tetrahedron T that is a fundamental region of one of above groups as follows.

Each bounding mirror r_i of T spans a hyperplane H_i going through the origin for $i = 1, 2, 3, 4$. After

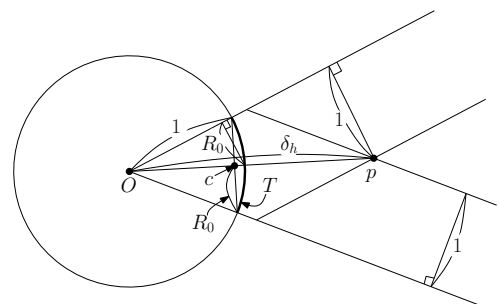


Figure 2.2: A two-dimensional analogue of finding the radius R_0 of an inscribed 0-sphere in a spherical 1-simplex T .

translating each H_i in the orthogonal direction of H_i toward the interior of the fundamental region by distance 1, let p be the intersection point of these translated hyperplanes. This point has equal distance 1 from all hyperplanes, but it does not lie on the unit sphere. Let δ_h be the distance of p from the origin O . Then, by rescaling the segment \overline{Op} by $1/\delta_h$ we get the center c of the inscribed sphere, and the Euclidean radius of T is $R_0 = 1/\delta_h$, see Figure 2.2. The approximated values of the Euclidean radii R_0 of fundamental regions of 4-dimensional finite Coxeter groups are computed in this way. These values are given in Table 2.1.

The smallest Euclidean radius occurs for G_4 and its value is $R_0 \geq 0.039102328$. Thus, the minimum distance $2R_0 \geq 0.0782$ as it is claimed in the following lemma.

Lemma 2.14. *Let G be a subgraph of a closest-pair graph of a point set in 4-space such that G contains no zero-degree vertices. If every component of the graph G is the orbit of a point under a four-dimensional symmetry group which is not $D_2^p \times D_2^q$, then the minimum distance δ is at least $0.07 > \delta_0$.*

Proof. For each fundamental region T , we consider all possibilities how the orbit of a point $u_0 \in T$ might give rise to a component of G . We have to place u_0 on some facets F of T and equidistant from the remaining facets of T . If the point is not chosen equidistant from the other faces, the closest-pair graph would not contain edges that generate all four mirrors. The minimum distance is achieved when u_0 lies in the interior of T and $F = \emptyset$. We analyzed each of the eight groups case by case in Table 2.1. This minimum distance $\delta = 2R_0 \approx 0.0782$ occurs when u_0 lies in the center of a barycentric subdivision of the facets of a 120-cell or of its dual, a 600-cell among the eight groups. \square

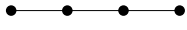
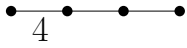
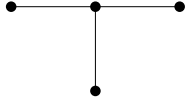
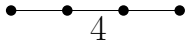
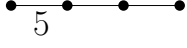

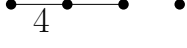
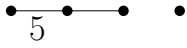
| Groups | Coxeter Diagrams | Normal Vectors | R_0 |
|------------------|---|---|---------------|
| A_4 |  | $(1, 0, 0, 0)$ $(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0)$ $(0, -\frac{1}{\sqrt{3}}, \frac{\sqrt{2}}{\sqrt{3}}, 0)$ $(0, 0, -\frac{\sqrt{3}}{2\sqrt{2}}, \frac{\sqrt{5}}{2\sqrt{2}})$ | 0.2236067977 |
| C_4 |  | $(1, 0, 0, 0)$ $(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0)$ $(0, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ $(0, 0, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ | 0.1429000737 |
| B_4 |  | $(1, 0, 0, 0)$ $(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0)$ $(0, -\frac{1}{\sqrt{3}}, \frac{\sqrt{2}}{\sqrt{3}}, 0)$ $(0, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{2}})$ | 0.1889822365 |
| F_4 |  | $(1, 0, 0, 0)$ $(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0)$ $(0, -\frac{\sqrt{2}}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0)$ $(0, 0, -\frac{\sqrt{3}}{2}, \frac{1}{2})$ | 0.09671356812 |
| G_4 |  | $(1, 0, 0, 0)$ $(-\frac{1+\sqrt{5}}{4}, \frac{\sqrt{10-2\sqrt{5}}}{4}, 0, 0)$ $(0, -\frac{2}{\sqrt{10-2\sqrt{5}}}, \frac{\sqrt{6-2\sqrt{5}}}{\sqrt{10-2\sqrt{5}}}, 0)$ $(0, 0, -\frac{\sqrt{10-2\sqrt{5}}}{2\sqrt{6-2\sqrt{5}}}, \frac{\sqrt{14-6\sqrt{5}}}{2\sqrt{6-2\sqrt{5}}})$ | 0.03910328003 |
| $A_3 \times A_1$ |  | $(1, 0, 0, 0)$ $(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0, 0)$ $(0, -\frac{1}{\sqrt{3}}, \frac{\sqrt{2}}{\sqrt{3}}, 0)$ $(0, 0, 0, 1)$ | 0.3015113445 |
| $C_3 \times A_1$ |  | $(1, 0, 0, 0)$ $(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0)$ $(0, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ $(0, 0, 0, 1)$ | 0.2108874992 |
| $G_3 \times A_1$ |  | $(1, 0, 0, 0)$ $(-\frac{1+\sqrt{5}}{4}, \frac{\sqrt{10-2\sqrt{5}}}{4}, 0, 0)$ $(0, -\frac{2}{\sqrt{10-2\sqrt{5}}}, \frac{\sqrt{6-2\sqrt{5}}}{\sqrt{10-2\sqrt{5}}}, 0)$ $(0, 0, 0, 1)$ | 0.1303737577 |

Table 2.1: Coxeter diagrams, outward unit normal vectors of fundamental regions, and inradii R_0 of fundamental regions of four-dimensional Coxeter groups.

Auxiliary Tools

3.1 Useful Principles

3.1.1 The Pruning and Condensing Principles

The pruning principle plays one of the most important roles in the new algorithm.

A general scheme of *conventional pruning* is as follows.

1. Classify a set of points by the value of some function F on a point.
2. Take only one class of points of the same value of F , preferably the smallest class, into consideration and *temporarily* ignore other sets.

We need to make sure that this pruning is canonical, i. e., the function F is equivariant under rotations; a function F of $x \in A$ is equivariant under a rotation R if $F(R \cdot x) = R \cdot F(x)$.

For example, given point sets A and B , if there is a point of non-zero distance to the origin in A , we can prune A and B by their distances to the origin as follows. First, classify points by their distances to the origin, count the number of points in each class, and temporarily focus only on the classes of the smallest cardinality A_s, B_s . After pruning by distances to the origin, we may assume that all the points are in the same distance to the origin, which means that they are on the same sphere. This is one of the main advantages of pruning. We can assume more structure on the point set after pruning.

Another advantage of pruning is that it reduces the number of points that we need to consider. If we can guarantee a size reduction by a factor $c < 1$, we say that we reduced the number of points successfully. Then, pruning can be repeated until it gets stuck, without affecting the time and space complexity. After pruning, we can restart the algorithm from the beginning with the pruned set without affecting the complexity of the algorithm, since $O(n \log n) + O(cn \log cn) + O(c^2n \log c^2n) + \dots = O(n \log n)$.

To assure that each pruning reduces the number of points by at least a half, we choose the set of the smallest cardinality. To ensure that the results of A and B are identical, we use some lexicographic rule for tie-breaking. We will only say that we “prune by a function F ”, without explicitly mentioning a tie-breaking rule.

We generalize this principle to a more general method, called *condensing* while maintaining the advantages of pruning: We *condense* a finite set A to a nonempty set $A' = F(A)$ of smaller size, not necessarily a subset of A , by using an equivariant function F , that is, $F(R \cdot A) = R \cdot F(A)$ for any rotation matrix R , where F is a function defined for any finite point set. The condensing principle can be applied not only to a point set but also to sets of tuples of points by defining a function F for tuples of points. For example, when we have a perfect matching of the points A that has been constructed by geometric functions, we can replace the set of edges by the midpoints of the edges. Afterwards we only consider these midpoints instead of the vertex set temporarily.

Any rotational invariants, such as distances and angles, can be used as equivariant functions

for condensing. The new algorithm also uses functions that are not rotational invariants but equivariant mappings: canonical axes (see Section 3.2.1) and edge figures (see Section 3.2.2) for example.

At first glance, condensing and pruning look dangerous because it *throws away information*. It could introduce new symmetries, and it might happen that the condensed/pruned sets are congruent, whereas the original sets are not. Therefore, the condensed/pruned set should be kept only temporarily. The prime goal of iterative condensing steps is to reduce point sets eventually to small enough sets A'' and B'' so that we can afford *dimension reduction*. Eventually we want to use dimension reduction techniques either in Section 3.3.1 or Section 5.5.

Pruning and condensing are very powerful and versatile methods, because we can use them with any equivariant functions or construction that one might think of as long as it is not too hard to compute. They allow us to concentrate on the cases where condensing makes no progress, and these cases are highly structured and symmetric. The difficulty is to pick the right pruning/condensing functions, and to decide how to proceed when pruning and condensing gets stuck.

3.1.2 Packing Arguments

Packing arguments are used throughout the new algorithm, when we know that the minimum distance between points or the minimum Plücker distance between planes is bounded from below and we want to show that the number of points or planes is bounded from above.

By using volume packing, we can easily establish the following lemma.

Lemma 3.1. *If there are m points on an unit d -sphere \mathbb{S}^d with the minimum distance δ then*

$$m \cdot \Omega_d(\delta/2) \leq \omega_d$$

where $\Omega_d(\delta/2)$ is the volume of the Euclidean d -ball of radius $\delta/2$ and ω_d is the surface area of a unit d -sphere \mathbb{S}^d .

Proof. Those m points can be replaced with disjoint open Euclidean d -balls of radius $\delta/2$. This forms a packing of such balls on the surface of the unit d -sphere. Thus, the sum of volumes of all those balls is smaller than the surface area of a unit d -sphere. \square

Also, we use the “kissing number” in various dimensions. A kissing number is the maximum number of non-overlapping unit balls each of which touches another given unit ball. The kissing number is known to be 6 in the plane, 12 in 3-space, 24 in 4-space, and between 40 and 44 in 5-space. In particular, the number of closest pairs, i.e., a pair whose distance is the minimum among all pairs in a given point set, that share the same point p is bounded by the kissing number.

The following constants are required in the new algorithm:

- K_5 is the kissing number on the surface of an unit 5-sphere \mathbb{S}^5 ; that is, the maximum number of non-overlapping unit 5-balls each of which touches another given unit 5-ball in a surface of \mathbb{S}^5 . Since the kissing number of 5-space is between 40 and 44, this number is less than or equal to 44. By defining K_2 similarly on a circle, it is known that $K_2 < 6$. Instead, we use the number 5 directly for the upper bound. We often use the kissing number 12 in 3-plane as well.
- $\delta_0 := 0.0005$ is the constant for applying the 1+3 dimension reduction. If the minimum distance of a point set is greater than δ_0 , the size of the point set is at most n_0 , where n_0 is the maximum packing of 3-balls of radius $\delta_0/2$ on a unit 3-sphere \mathbb{S}^3 . Then, we can apply the dimension reduction principle in Section 3.3.1 without affecting the time complexity.

Otherwise, we need to go through other condensing procedures. The value of δ_0 will be justified in Lemma 5.2 in Section 5.2.

Since the surface area ω_3 of \mathbb{S}^3 is $2\pi^2$ and the volume $\Omega_3(\delta_0/2)$ of a 3-ball of radius $\delta_0/2$ is $\frac{4}{3}\pi(\delta_0/2)^3$, by Lemma 3.1, a rough estimate of the upper bound of n_0 is

$$n_0 \leq 2\pi^2 / (\frac{4}{3}\pi(\delta_0/2)^3) < 3.016 \times 10^{11}. \quad (3.1)$$

- C_1 denotes the maximum number of planes at pairwise Plücker distance greater than or equal to δ_{\min} . This bound is used will be used in Algorithm M. Here δ_{\min} is defined as follows. The Plücker distance of two isoclinic planes P, Q is δ_{\min} if P, Q are mapped to two adjacent vertices of an icosahedron by a Hopf fibration. By Lemma 2.3, the Plücker distance between two planes of angle α , α is $\sqrt{2} \sin \alpha$ and by Lemma 2.11, such two planes are mapped to a pair of points of geodesic distance 2α by the Hopf fibration. The edge length of an icosahedron with a circumscribed unit sphere is $\delta_{\text{ico}} = \sqrt{50 - 10\sqrt{5}}/5$ so the geodesic distance of two vertices in the sphere is

$$2 \arcsin(\frac{1}{2} \cdot \delta_{\text{ico}}) \approx 1.107148718.$$

Then the angle between two such planes is

$$\alpha_{\min} = \arcsin(\frac{1}{2} \cdot \delta_{\text{ico}}) \approx 0.5535743590.$$

The Plücker distance of such pair of planes is

$$\delta_{\min} = \sqrt{2} \sin(\arcsin(\frac{1}{2} \cdot \delta_{\text{ico}})) \approx 0.7434960688.$$

The Grassmannian $\mathbb{G}(2, \mathbb{R}^4)$ with Plücker distances can be regarded as a double cover of the unit 5-sphere. By Lemma 3.1, the upper bound can be computed as

$$C_1 \leq \pi^3 / (\frac{8}{15}\pi^2(\delta_{\min}/2)^5),$$

since the surface area ω_5 of a unit 5-hemisphere is π^3 and the volume $\Omega_5(\delta_{\min}/2)$ of a 5-ball of radius $\delta_{\min}/2$ is $\frac{8}{15}\pi^2(\delta_{\min}/2)^5$. Thus,

$$C_1 \leq 829. \quad (3.2)$$

This bound is a rough estimate and the correct bound is likely to be much smaller. Packing in Grassmannian spaces has already been considered by Conway, Hardin and Sloane [12]. They used the *chordal distance* defined as $\sqrt{\sin^2 \alpha + \sin^2 \beta}$ for a pair of planes at angle $\{\alpha, \beta\}$ to provide the experimental bounds. See before in Section 2.3 for comparisons with Plücker distance. According to their experimental results, for planes in 4-space, when there are 37 planes, the maximum chordal distance is at most approximately 0.728633689875024. Their results suggest that a smaller upper bound of C_1 is likely to exist.

3.2 Representation Details

3.2.1 Canonical Axes

The canonical-axes construction is a well-known procedure for detecting the rotational symmetries in a planar point configuration [28, 4, 20]. We use this construction in Step C11 of

Algorithm C in Section 5.1. We can encode labeled-point sets on a circle as a cyclic string. This can be done by alternating between labels for a point and angular distances between two adjacent points [28].

In addition, by making cyclic shifts in this string so that the string becomes lexicographically smallest, but still starts with labels (not angles), we can get a unique representation of points on a circle with labels. These starting points that yield the same string give rise to a set of p equidistant rays starting from the origin. We call a collection of these points *canonical axes*. Canonical axes can be found in $O(n \log n)$ time when n is the number of labeled points by standard string-processing techniques [27]. Canonical axes have the same rotational symmetries as the original configuration.

We assume labels are preserved under rotations. Then, if a circle K is mapped to a circle K' , canonical axes of K are mapped to canonical axes of K' , so canonical axes are equivariant under rotations.

Moreover, canonical axes are more special than most other equivariant functions. If canonical axes of a circle K are mapped to canonical axes of another circle K' by a rotation R , K' itself is also mapped to K by R , so the converse of equivariance is also true.

In this context, we define a canonical set and a canonical set procedure as a special condensing. Refer to Section 5.5 about a canonical set procedure. Canonical axes are the result of a canonical set procedure for a circle and rotational symmetries on a circle. The new algorithm employs a canonical set procedure for a square torus and translational symmetries on a torus. See Section 5.5.

3.2.2 Representation of Congruence Types

Assume that a geometric graph G is given. A vertex figure of a vertex v , that is, v together with its neighbors, is first introduced by Atkinson's algorithm [5] (refer 1.3.2). Atkinson also argued that a vertex figure can be encoded in a string of length $O(1)$ if the vertex is of bounded degree but he omitted the details about how to actually represent a vertex figure. The new algorithm uses vertex figures and also introduces an extended concept called an edge figure of an edge uv , that is, all neighbors of u and v together with the edge uv .

In this section, the representation details of a vertex figure, and of an edge figure are described. We assume that a given graph $G = (V, E)$ is a directed graph embedded in a 3-sphere \mathbb{S}^3 such that all the edge lengths are the same and G has its maximum degree at most 12. This is because we use vertex figures and edge figures for a directed copy of a closest-pair graph on a 3-sphere. The kissing number in 3-space is 12.

Let us begin with a vertex figure. We may consider a vertex v of degree greater than 2, because if a vertex v is of degree 1, there exists only one congruence type of the neighborhood of v and if a vertex v is of degree 2, the angle between two incident edges determines its congruence type.

Then, we can choose two distinct incident edges va and vb such that an angle $\angle(avb)$ attains the minimum among all pairs of such edges. We order a and b and call these two edges a base pair. Observe that the vectors vO (O is the origin), va , vb are not coplanar. Otherwise v, a, b are in a great circle, v has only a and b as its neighbors and v is of degree two.

Let n_1, n_2, n_3 be the vectors obtained by applying the Gram-Schmidt orthogonalization process to the ordered vectors vO, va, vb . Let n_4 be the unit normal vector to 3-space spanned by n_1, n_2, n_3 such that a basis n_1, n_2, n_3, n_4 is positively oriented. Four vectors n_1, n_2, n_3, n_4 can endow a Cartesian coordinate system with the new origin v . Sort all the coordinates of other neighbors of v lexicographically. Tag each coordinate with direction labels, -1 for outgoing edges and +1 for ingoing edges at the end of the coordinates. The concatenated string of coordinates in the increasing order is a corresponding string to the base pair va and vb . Obtain such strings

for all the base pairs and tag direction labels for the base pair. Sort those obtained strings. The lexicographically-smallest string is the congruence type of the vertex v , or the representation of the vertex figure of the vertex v .

The congruence type of a directed edge uv is defined similarly by using a Cartesian coordinate system endowed by some base pair. In this case, we choose the base pair in a way that the base pair always includes uv and another incident edge va in the base pair forms the minimum angle to uv .

For both vertex figures and edge figures, it takes time $O(c^2 \log c) = O(1)$ for construction and is of length $O(c) = O(1)$ where the maximum degree $c \leq 12$.

3.3 Leaping From Previous Ideas

3.3.1 1 + 3 dimension Reduction

In 4- or higher-dimensional space, the first approach by Alt et al. [2] is by *dimension reduction*. One of the dimension reduction methods for the new algorithm is more a variant of one in Akutsu [1], although Akutsu's method is also similar to the method by Alt et al. Their differences and details was described earlier in Section 1.3.3.

Our variant of the dimension reduction method is as follows. We pick an arbitrary point $a_0 \in A$. This point must be mapped to one of the points in B . For any $b \in B$, we try to rotate A so that a_0 lies on b . To look for rotations R that leave this point fixed, we project A and B on the hyperplane H perpendicular to $Oa_0 = Ob$. To each projected point, we attach the signed distance from H as a label. A rotation in H that preserves labels can be extended to a rotation R that fixes the point a_0 . For each $b \in B$, the problem is thus reduced to the one-lower dimensional congruence testing problem for labeled point sets.

The new algorithm uses this method for A and B in 4-space in the following situation. Suppose that we know pruned sets A_0, B_0 obtained by A and B such that $|A_0| \leq n_0$ and A_0 is mapped to B_0 by the pruning principle where n_0 is defined in (3.1). Then by limiting the choice of a_0 in A_0 and the choice of b in B_0 , we can check the congruence of A and B in time $O(n_0 n \log n) = O(n \log n)$ by known methods for 3-space.

3.3.2 Closest-pair Graphs

A closest-pair graph of a given point set P is a graph G such that each point in P is a vertex of G and a pair of vertices p and q in P forms an edge if and only if the distance between p and q achieves the minimum distance among all the pairs in P . A closest-pair graph is different from a nearest-neighbor graph, a directed graph each of whose edge from p to q indicates that q is a nearest-neighbor of p . In particular, a nearest-neighbor graph is a directed graph, since the nearest-neighbor relation is not symmetric and each vertex of a nearest-neighbor graph is of strictly-positive outdegree. On the other hand, a closest-pair graph is undirected and some vertices of a closest-pair graph may have zero degree.

Notable characteristics of a closest-pair graph are: (1) it can be constructed in time $O(n \log n)$ by divide and conquer [32, 6] in any fixed dimension, (2) the degree of each vertex is bounded by the "kissing number", as described earlier in Section 3.1.2, and (3) the number of edges is at most linear to the number of vertices because of (2).

By taking advantage of these characteristics, closest pairs were already used in previous work [5, 10]. See before in Section 1.3.2. The new algorithm computes a closest-pair graph of not only input point sets in 4-space but also sets of planes in Plücker space. See before in Section 2.3.

3.3.3 Finding Representative Points from a 2-sphere: Algorithm K

The following algorithm is a variant of Atkinson’s algorithm [5] (refer to Section 1.3.2) to condense a finite point set in a 2-sphere. Although Atkinson’s algorithm preserves rotational symmetries as a canonical set procedure (refer to Section 3.2.1 and Section 5.5.1), this algorithm is rather an efficient condensing method. The algorithm finds at most 12 representative points in \mathbb{S}^2 by condensing as in Lemma 3.2. This algorithm is used in Section 5.4 to condense great circles in the same Hopf bundle.

Lemma 3.2 (Algorithm K). *There is a procedure that reduces a set F of points on the 2-sphere \mathbb{S}^2 to a nonempty set F' of at most $\min\{12, |F|\}$ representative points on \mathbb{S}^2 , in $O(|F| \log |F|)$ time. These points are either the vertices of a regular tetrahedron, a regular octahedron, a regular icosahedron, a single point, or a pair of antipodal points. This mapping from sets F to sets F' is equivariant under rotations and reflections.*

Proof. We repeatedly prune the set F . We start by computing the convex hull of F . If it does not contain the origin, we can immediately output the vector pointing to the center of gravity of the points as a representative. If the hull is one-dimensional or two-dimensional, we get two antipodal points as representatives.

We can thus assume that the hull is a three-dimensional polytope. If the vertex degrees are not all the same, we prune the point set and restart. We can thus assume that the graph of the polytope is regular, and by Euler’s formula, the degree d can be 3, 4, or 5. Euler’s formula also yields the number f of faces in terms of the number n of vertices: $f = (d - 2)/2 \cdot n + 2 \leq \frac{3}{2}n + 2$. We now try to prune by face degrees. If there are different face degrees, we replace F by the centroids of the smallest class of faces, and restart. We have condensed $|F|$ at least by a factor of $3/4$ (the additive term $+2$ is negligible as long as $|F|$ is large). The remaining case is when all face degrees and all vertex degrees are equal. Then the polytope must have the combinatorics of one of the five regular polytopes: the tetrahedron, the octahedron, the icosahedron, the cube, or the dodecahedron. For the cube and the dodecahedron, we take the centroids of the faces and thereby reduce the number of points.

We are left with the case of the tetrahedron, the octahedron, and the icosahedron. If the edges don’t have the same lengths, we condense and replace F by the midpoints of the smallest edge class. This will lead to a reduction in the number of points, except for an octahedron and icosahedron with two edge lengths, each occurring at least 6 times or 12 times, respectively. We claim that the triangular faces cannot be all congruent in this case, and we can prune the 8, resp. 20, faces by congruence type, leading to a reduction to at most 4 resp. 10 triangle centroids. If the triangles were all congruent, their sides would all have to be long-long-short or long-short-short. This would mean, for the octahedron, that the 12 edges’ lengths are divided in the proportion 4 : 8, and in that case we could have condensed by the edges. The same argument works for the icosahedron, because the 30 edges’ lengths are divided in the proportion 10 : 20.

Thus, the only cases where pruning cannot proceed are a tetrahedron, or an octahedron, or an icosahedron made of regular triangles. These must be the regular polytopes.

The most expensive part in each pruning step is the computation of the convex hull, which takes $O(n' \log n')$ time, where n' is the size of the current point set. Each step reduces n' by a constant factor of at most $11/12$. Thus the overall running time is $O(|F| \log |F|)$. \square

II

The New Algorithm in 4-space

Finally, the new optimal algorithm in 4-space is introduced in this part. Chapter 4 describes the flow of the new algorithm and how all the modules are related.

As can be seen in Figure 4.1, the new algorithm consists of six main modules; iterative pruning, generating orbit cycles, marking and pruning great circles, the mirror case, 2+2 dimension reduction and 1+3 dimension reduction. The module called “1+3 dimension reduction” is already well explored in Section 3.3.1. We explain the rest of other five modules in Chapter 5. Each section of Chapter 5 covers one of the remaining five modules.

It is likely that our algorithm can be simplified. We also believe that the constants in the bounds are certainly larger than the actual bounds.

This algorithm was published in [26] by joint work with Günter Rote.

Overview of the New Algorithm

Preliminaries. For packing arguments, we denote by K_d the kissing number on \mathbb{S}^d : the maximum number of equal interior-disjoint balls on the d -dimensional sphere \mathbb{S}^d that can simultaneously touch a ball of the same size. We use the known bounds $K_2 = 5$, $K_3 = 12$, and $40 \leq K_5 \leq 44$.

We will declare the problem to be trivially solvable if the closest-pair distance δ is large, i.e., $\delta > \delta_0 := 0.0005$. This implies that the input size $|A|$ is bounded by $n_0 < 3.016 \times 10^{11}$, and hence, by 1+3 dimension reduction, the problem can be reduced to at most n_0 instances of 3-dimensional congruence testing, taking $O(n \log n)$ time overall.

Remark. Whenever we apply some procedure, e.g., condensing, on A , we will apply the same procedure to the other set B in parallel but we will mostly describe those steps only for the set A . If A and B are congruent, B will undergo exactly the same sequence of steps as A . If a difference manifests itself at any point, we know that A and B are not congruent, and we can terminate.

General Flow. We first give a rough overview of our algorithm, omitting details and some special cases. See Fig. 4.1.

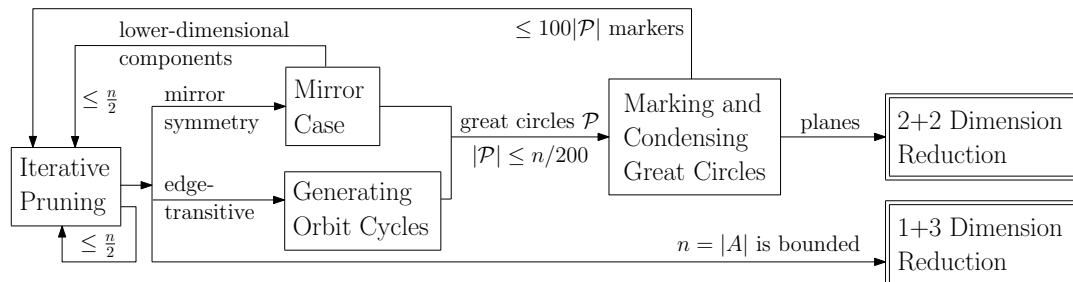


Figure 4.1: The general flow of the algorithm.

Our goal is to apply the condensing principle repetitively until we can apply either of the two dimension reduction principles to the original input point sets (not to the pruned sets): 1+3 dimension reduction (see Section 3.3.1) or 2+2 dimension reduction (see Section 5.5). The 1+3 dimension reduction makes a line and the orthogonal 3-space invariant, and the 2+2 dimension reduction makes two orthogonal 2-planes invariant.

Whenever the number of pruned points is smaller than some chosen threshold n_0 , we can afford 1+3 dimension reduction as mentioned in the preliminaries. If we find a set \mathcal{P} of equivariant great circles from A and B , such that $1 \leq |\mathcal{P}| \leq C_1$, where C_1 is defined in Section 3.1.2, we can trigger 2+2 dimension reduction.

By applying the 2+2 dimension reduction technique described in Section 5.5, the problem boils down to congruence testing of “labeled” points on a two-dimensional torus under trans-

lations. This problem can be solved in $O(n \log n)$ by reducing a point set but still preserving symmetries and by using the Voronoi regions as the neighborhood structure.

The congruence testing algorithm begins with pruning by distance from the origin. We may thus assume without loss of generality that the resulting set A' lies on the unit 3-sphere $\mathbb{S}^3 \subset \mathbb{R}^4$.

In iterative pruning (Section 5.1), we first check if the minimum distance δ between points of A' is bigger than the threshold δ_0 . If yes, we conclude that $|A'| \leq n_0$ and trigger 1+3 dimension reduction. Otherwise, we construct the closest-pair graph G on A' . Then we prune the edges of G by congruence type of its edge figures. An edge figure consists of two adjacent vertices and all their neighbors. We apply pruning with other criteria to the set A' until the resulting set A_0 cannot be further reduced. Then, all edge neighborhoods in the graph G are congruent. This allows us to find either *orbit cycles* in G , or a subset of A_0 with mirror symmetry. An orbit cycle is a cyclic path $a_1 a_2 \dots a_\ell a_1$ with a rotation R such that $R a_i = a_{(i \bmod \ell) + 1}$; in other words, it is the orbit of the point a_1 under a rotation R .

Mirror symmetry is the symmetry that, for each edge of G , swaps the two endpoints and maps the whole point set onto itself. It implies that the point set A_0 must be related to one of the regular four-dimensional polyhedra, or A_0 (and G) is the Cartesian product of two regular polygons in orthogonal planes. This follows from the classical classification of discrete reflection groups. The former case can be excluded, since $\delta < \delta_0$, and in the latter case, we can proceed to Section 5.4 for marking and condensing great circles. See Section 5.3.

Let us look at the case when we have found orbit cycles (Section 5.2) or a set of pairs of regular polygons in orthogonal planes (Section 5.3). Geometrically, an orbit cycle lies on a helix around a great circle C and a pair of orthogonal planes intersect a sphere as a pair C, C' of great circles. Thus, we get a collection \mathcal{P} of great circles on \mathbb{S}^3 . We treat these great circles as objects in their own right, and we construct the closest-pair graph on \mathcal{P} . For this, we use the Plücker embedding of the corresponding planes into $\mathbb{S}^5/\mathbb{Z}_2 = \mathbb{RP}^5$, mapping each plane to a pair of antipodal points on the 5-sphere \mathbb{S}^5 . Then, we construct a closest-pair graph of planes with respect to the Plücker distance, defined as a normalized Euclidean distance in the Plücker embedding. Refer to Section 2.3 for the Plücker embedding and Plücker distance.

For each closest pair (C, D) of great circles in \mathcal{P} , if they are not isoclinic (i.e., the projection of C to the plane containing D is an ellipse but not a circle) the major axis of this ellipse marks two points of C as described in Paragraph “Equivalent Definition by Orthogonal Projections” in Section 2.2. The set of all markers replaces the set A_0 . This completes a successful condensing step, and we restart and continue pruning as before.

Otherwise, all projected “ellipses” turn out to be circles. In this case, we can find a subfamily of great circles in a special position: they must be part of a *Hopf bundle* of circles. The circles of a Hopf bundle can be mapped one-to-one to points on the 2-sphere \mathbb{S}^2 . See Section 2.4. We can thus use the condensing procedures for a 2-sphere in three dimensions as in Section 3.3.3. This yields a small set \mathcal{P} of at most 12 great circles. Then, we can apply the 2+2 dimension reduction technique.

The details for marking or condensing great circles are actually more complicated, since we might have a phase in which \mathcal{P} is successively pruned, see Section 5.4.

This concludes the summary of the algorithm. The steps of the algorithm involve several different operations: We need closest-pair graphs in 4 and 6 dimensions. The closest-pair graph of k points can be calculated in $O(k \log k)$ time in any fixed dimension, by a classical divide-and-conquer approach [6]. We also compare a pair of edge figures in \mathbb{S}^3 but this takes only time $O(1)$ because the maximum degree is at most 12. We also need Voronoi diagrams in two dimensions and convex hulls in three dimensions. Finally, we need to sort lists of numbers lexicographically. In summary, we will be able to reduce the size of the current point set A' by a constant factor less than $\frac{1}{2}$, in $O(|A'| \log |A'|)$ time, until dimension reduction is possible.

Hence, we obtain the following theorem.

Theorem 4.1. *We can decide if two n -point sets A and B in 4-space are congruent in $O(n \log n)$ time and $O(n)$ space.*

Five Crucial Modules

5.1 Iterative Condensing Based on the Closest-Pair Graph: Algorithm C

After pruning by the distance to the origin, we have a set $A \subset \mathbb{R}^4$ of n points with equal distance from the origin. This set may already be the result of some previous condensing steps. Without loss of generality, we may assume to be given a set A on the unit sphere \mathbb{S}^3 . We construct the closest-pair graph G for a point set A on the 3-sphere and try to condense it.

In this section, we perform an algorithm, which is a sequence of pruning and condensing to obtain a new equivariant point set A' that is in one of the following cases:

- (i) the closest distance δ of A' is greater than δ_0 , or
- (ii) the closest-pair graph of A' has mirror symmetry, or
- (iii) the closest-pair graph of A' is in the edge-transitive case.

If (i) happens, we apply 1+3 dimension reduction. If (ii) or (iii) happens, we proceed to Algorithm M in Section 5.3 or Algorithm O in Section 5.2 respectively. Those sections explain how to reduce the cases to 2+2 dimension reduction.

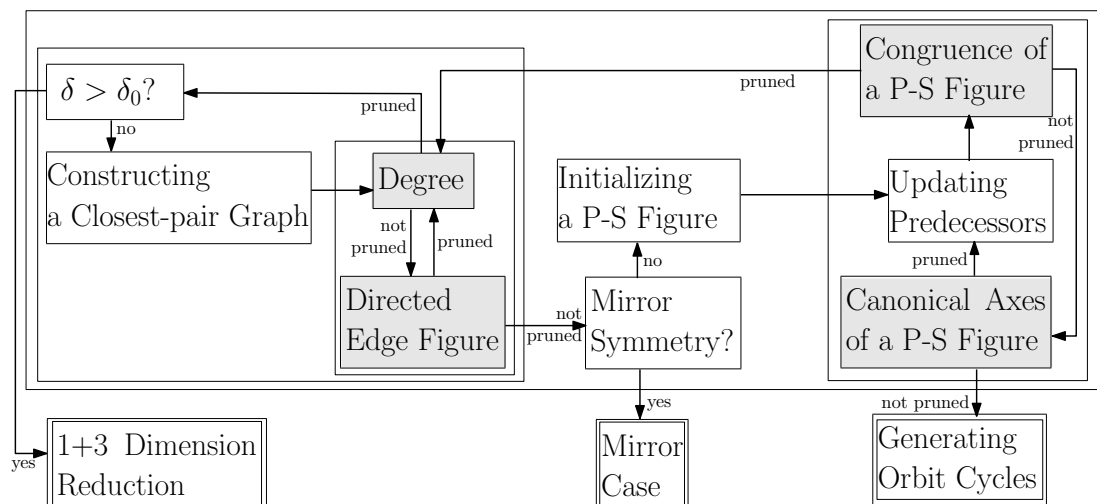


Figure 5.1: Iterative pruning. The shaded boxes represent pruning/condensing steps with pruning criteria or condensing methods.

We first explain the notion of a *predecessor-successor figure* in G . We have a directed edge uv of G together with a set of *predecessor edges* $p(uv)$ incident to u and a set of *successor edges* $s(uv)$ incident to v , as in Fig. 5.2a. All edges have the same length, their endpoints lie on the 3-sphere \mathbb{S}^3 , and all predecessor and successor edges form the same angle α with uv . Then the

endpoints of these edges lie on two circles. If we reflect the predecessor circle at the bisecting hyperplane of uv , it comes to lie on the successor circle. This results in one circle with a succinct representation of the geometric situation. See Fig. 5.2a and Fig. 5.2b. We refer to the endpoints of the predecessor and successor edges as *predecessors* and *successors*.

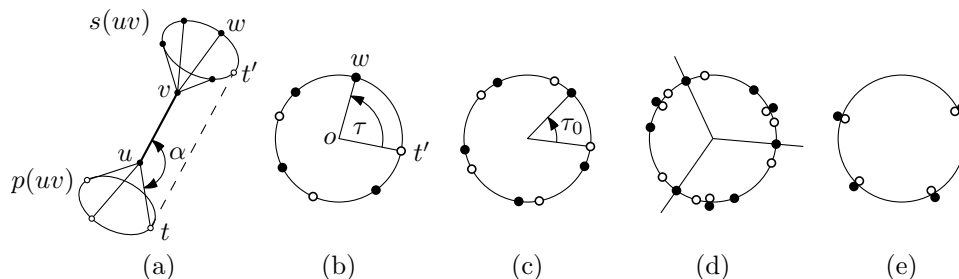


Figure 5.2: (a) Predecessors and successors of uv at angle α ; t' is the reflected copy of t on the successor circle. (b) The corresponding predecessor-successor figure, and the torsion angle $\tau(tuvv)$. Predecessors are drawn white and successors black. (c) An edge-transitive predecessor-successor figure with torsion angle τ_0 . (d) Canonical axes. (e) Mirror symmetry.

The following algorithm guarantees that the outcome A' is in one of the promised cases.

Algorithm C (*Prune the closest-pair graph*). We are given a set $A \subset \mathbb{R}^4$ of n points, equidistant from the origin. This set may already be the result of some previous condensing steps. Without loss of generality, we may assume that A lies on the unit sphere \mathbb{S}^3 .

- C1.** [Well-separated points?] Compute the closest distance δ between points of A . If $\delta > \delta_0 = 0.0005$, apply 1+3 dimension reduction. ($|A|$ is bounded by a constant.)
- C2.** [Construct the closest-pair graph.] Construct the closest-pair graph G , and initialize the directed graph D with two opposite arcs for every edge of G . (This takes $O(n \log n)$ time, and the degrees in G are bounded by $K_3 = 12$.)
- C3.** [Prune by degree.] If the indegrees and outdegrees in D are not all the same, prune the vertices by degree, and return to C1, with the smallest class A' taking the role of A . (Otherwise, we now enter a loop in which we try to prune arcs from D .)
- C4.** [Prune by directed edge figure.] The *directed edge figure* of an arc $uv \in D$ consists of uv together with all arcs of D out of v and all arcs of D that lead into u . If the directed edge figures are not all congruent, *prune the arcs of D* by congruence type of directed edge figures, and return to C3. (Here we apply the pruning principle not to points but to arcs and we replace the edge set D by a smaller subset D' . Since the degrees are bounded, we can compare two directed edge figures in constant time.)
- C5.** [Mirror symmetry?] (Now all arcs have the same directed edge figure.) If the directed edge figure of uv is symmetric with respect to the bisecting hyperplane of uv , proceed to Algorithm R (Section 5.3).
- C6.** [Choose an angle α with no mirror symmetry.] Pick an angle α for which the predecessors and successors are not completely symmetric, that is, the predecessor-successor figure does not look like Fig. 5.2e.
- C7.** [Initialize Successors.] For every arc $uv \in D$, set $s(uv) := \{vw : vw \in D, \angle uvw = \alpha\}$. (The size of $s(uv)$ is bounded by $K_2 = 5$. We will now enter an inner loop in which we try to prune arcs from the sets $s(uv)$.)

- C8.** [Update predecessors.] Define the predecessor edges by $p(uv) := \{tu : uv \in s(tu)\}$. Build the predecessor-successor figure for each arc, as explained in the text above.
- C9.** [Prune by predecessor-successor figures.] If there are arcs whose predecessor-successor figures are not congruent, prune the arcs of D accordingly, and return to C3.
- C10.** [Check regularity.] (Now all arcs have the same predecessor-successor figure. Each figure must contain the same number k of predecessors and successors, since the total number of predecessors in all figures must balance the total number of successors.) If the predecessor-successor figure consists of two regular k -gons, proceed to Algorithm O for generating orbit cycles, see Section 5.2. (We call this the *edge-transitive* case. See Figure 5.2c.)
- C11.** [Prune successors by canonical axes.] Prune $s(uv)$ to a proper nonempty subset by computing *canonical axes* as explained in Section 3.2.1, and return to C8.

An example of canonical axes is depicted in Fig. 5.2d. If canonical axes consist of p axes, we know that $p < k$ because the maximally symmetric case of two regular k -gons (the *edge-transitive* case shown in Fig. 5.2c) has been excluded in Step C10. The loop from C8–C11 maintains the following loop invariant on entry to Step C10:

$$\textit{There is a position of a successor that is not occupied by a predecessor.} \quad (5.1)$$

For the reduction in Step C11, we rotate the canonical axes counterclockwise until they hit the first position of type (5.1). The successors that are intersected by the canonical axes form a nonempty proper subset $s' \subseteq s(uv)$. We thus replace $s(uv)$ for each edge uv by s' and return to Step C8. By construction, we have made sure that (5.1) still holds. After pruning all successor sets, the predecessor sets are reduced accordingly in Step C8, but this cannot invalidate (5.1). (The invariant (5.1) holds on first entry to the loop because of Step C6.)

The algorithm has three nested loops (indicated by indentation) and works its way up to higher and higher orders of regularity. After C3, all vertices have the same degree. After C4, we know that all *pairs* of adjacent vertices look the same. If we exit to Algorithm O in Step C10, we will see that certain chains of *four* points can be found everywhere.

There is the *global loop* that leads back to C1 after each successful pruning of vertices by degree. Since the size of A is reduced to less than a half, we need not count the iterations of this loop. In addition, there is an outer loop that resumes working at C3 after pruning the edges of D , and an inner loop that starts at C8 and is repeated whenever the successor set $s(uv)$ is pruned. In these loops, we maintain that $D \neq \emptyset$ and $s(uv) \neq \emptyset$. In Step C3, if we have removed at least one edge from D , we will either be able to prune by degree, or the degree of all vertices has gone down by at least one. Since the degree is initially bounded by 12, Step C3 can be visited at most 12 times before exiting to C1. Similarly, Step C11 removes at least one element of $s(uv)$, so this loop is repeated at most 5 times before there is an exit to the outer loop in step C9. The most time-consuming step is the construction of the closest-pair graph in Step C2. All other operations take $O(n)$ time, not counting the exits to Algorithms M and O. Thus, the overall time is $O(|A| \log |A|)$.

5.2 Generating Orbit-Cycles: Algorithm O

We now describe how, in the edge-transitive case, the algorithm O produces a set \mathcal{P} of at most $|A|/200$ great circles. All predecessor-successor figures look like Fig. 5.2c. The *torsion angle* $\tau(tuvw)$ between a predecessor edge $tu \in p(uv)$ and a successor edge $vw \in s(uv)$ is the oriented angle $\angle(t'ow)$ in the predecessor-successor figure of uv . We define τ_0 as the smallest counterclockwise torsion angle that appears in the predecessor-successor figure. Let $t_0u_0v_0w_0$ be a fixed quadruple with this torsion angle.

- Lemma 5.1.**
1. For every $a_2a_3 \in s(a_1a_2)$, there is a (unique) edge $a_3a_4 \in s(a_2a_3)$ such that $a_1a_2a_3a_4$ is congruent to $t_0u_0v_0w_0$.
 2. Moreover, there is a unique rotation R_0 that maps $t_0u_0v_0w_0$ to $a_1a_2a_3a_4$.
 3. For every triple $a_1a_2a_3$ with $a_2a_3 \in s(a_1a_2)$, there is a unique cyclic sequence $a_1a_2 \dots a_\ell$ such that $a_i a_{i+1} a_{i+2} a_{i+3}$ is congruent to $t_0u_0v_0w_0$ for all i . (Indices are taken modulo ℓ .)
 4. Moreover, there is a unique 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 .
 5. The points $a_1a_2 \dots a_\ell$ do not lie on a circle.

Proof. We first establish two facts about $t_0u_0v_0w_0$.

$$\text{The four points } t_0, u_0, v_0, w_0 \text{ do not lie on a circle.} \quad (5.2)$$

$$\text{The three points } t_0, u_0, v_0 \text{ do not lie on a great circle.} \quad (5.3)$$

If t_0, u_0, v_0, w_0 would lie on a circle (not necessarily through the origin), the predecessor-successor figure of u_0v_0 would have a mirror-symmetric predecessor $t_0u_0 \in p(u_0v_0)$ and successor $v_0w_0 \in s(u_0v_0)$, in contradiction to the choice of α in Step C6 and to the invariant (5.1). If the points t_0, u_0, v_0 lie on a great circle C , then w_0 would also lie on C , since $\angle t_0u_0v_0 = \angle u_0v_0w_0 = \alpha$, but this would contradict (5.2).

Now, let a_1, a_2, a_3 be any three points with $a_2a_3 \in s(a_1a_2)$. By the definition of predecessors, $a_1a_2 \in p(a_2a_3)$. We can thus fit $t_0u_0v_0$ to $a_1a_2a_3$ in the predecessor-successor figure of a_2a_3 . Since the points t_0, u_0, v_0 are not on a great circle, they span a three-dimensional subspace, and the rotation R_0 that maps $t_0u_0v_0$ to $a_1a_2a_3$ is uniquely determined. Since all predecessor-successor figures are congruent, this means that the edge v_0w_0 is mapped to some successor edge $a_3a_4 \in s(a_2a_3)$. This establishes Properties 1, 2, and 5.

This process can be continued: Since $a_2a_3 \in p(a_3a_4)$, there is a unique edge $a_4a_5 \in s(a_3a_4)$ such that $a_2a_3a_4a_5$ is congruent to $t_0u_0v_0w_0$, and so on.

By Property 2, there are two unique rotations from $t_0u_0v_0w_0$ to $a_1a_2a_3a_4$ and to $a_2a_3a_4a_5$, and thus there is a unique rotation R' with $R'[a_1a_2a_3a_4] = [a_2a_3a_4a_5]$. We have seen that the points $a_1a_2a_3$ are not on a great circle, being congruent to $t_0u_0v_0$ and thus the rotation R is already uniquely specified by the conditions $R[a_1a_2a_3] = [a_2a_3a_4]$. We have therefore established that $a_{i+1} = Ra_i$ for $i = 1, 2, 3$ implies $a_{i+1} = Ra_i$ for $i = 4$. This can be continued by induction, implying that $a_1a_2 \dots$ is the orbit of the point a_1 under the rotation R .

Since the number of points in A is finite, this orbit must return to a_1 and form a directed cyclic path $a_1a_2 \dots a_\ell$ in the closest-pair graph D . This establishes Properties 3 and 4. \square

We call the cyclic paths that are constructed in Lemma 5.1 *orbit cycles*. The following lemma gives a bound on the number of orbit cycles in terms of $|A|$ when the closest-pair distance is small enough.

Lemma 5.2. *The number of orbit cycles is at most $|A|/200$ provided that the closest distance δ is smaller than $\delta_0 = 0.0005$.*

Proof. We have $a_{i+1} = R_{\varphi,\psi}a_i$ for all i with an appropriate basis $x_1y_1x_2y_2$ where $R_{\varphi,\psi}$ is defined as (2.1). We cannot have $\varphi = 0$ or $\psi = 0$, because otherwise the orbit would form a regular polygon $a_1a_2a_3a_4 \dots a_\ell$ in a plane, contradicting Lemma 5.1.5. Thus, we know that $\varphi, \psi \neq 0$. To get a closed loop, we must have $|\varphi|, |\psi| \geq 2\pi/\ell$. If the projection of a_i to the x_1y_1 -plane has norm r_1 and the projection to the x_2y_2 -plane has norm r_2 with $r_1^2 + r_2^2 = 1$, then the squared distance is

$$\delta^2 = \|a_{i+1} - a_i\|^2 = (2r_1 \sin \frac{|\varphi|}{2})^2 + (2r_2 \sin \frac{|\psi|}{2})^2 \geq 4 \sin^2(\pi/\ell).$$

As a result,

$$\delta \geq 2 \sin \frac{\pi}{\ell} \tag{5.4}$$

Note that Fenchel's theorem [18] yields the same bound. Thus, if $\delta \leq 2 \sin(\pi/12000) \approx 0.000523$, it is guaranteed that $\ell \geq 12000$, that is, every orbit cycle contains at least 12000 points. On the other hand, each point $u \in A$ belongs to a bounded number of orbit cycles: it has at most $K_3 = 12$ incoming arcs tu , and each each arc tu has at most $K_2 = 5$ successor edges $uv \in s(tu)$. The triple tuv specifies a unique orbit cycle, and thus there are at most $12 \times 5 = 60$ orbit cycles through u . The total number of orbit cycles is therefore at most $|A| \cdot 60/12000 = |A|/200$. \square

For each orbit cycle, we can find an appropriate rotation matrix $R_{\varphi,\psi}$. If $\varphi = \pm\psi$, then the orbit of any point under this rotation lies on a great circle, contradicting Lemma 5.1.5. Thus, we get a unique invariant plane that rotates by the smaller angle in itself. We replace each orbit cycle by the great circle in this invariant plane, yielding the desired set \mathcal{P} of great circles with $|\mathcal{P}| \leq |A|/200$.

The same computation used in Lemma 5.2 can be used in Section 5.3 to give a bound on the number of points in a regular polygon inscribed in a great circle. It is stated as the following corollary.

Corollary 5.3. *If the closest distance $\delta < \delta_0$ and there is a regular k -gon inscribed in a great circle of a sphere, then $k > 12000$.*

Proof. The same result as Lemma 5.2 holds for a great circle, but not when $\varphi = 0$ or $\psi = 0$. From (5.4) and $2 \sin(\pi/12000) > \delta_0$, we find that $k > 12000$. \square

5.3 The Mirror Case: Algorithm R

Algorithm R (*Treat mirror-symmetric closest-pair graphs*). We are given a nonempty directed subgraph D of the closest-pair graph on a point set $A \subset \mathbb{S}^3$ with closest-pair distance $\delta \leq \delta_0$. All directed edges in D have equal edge-figures and exhibit perfect mirror-symmetry, as established in Steps C4 and C5 of Algorithm C. Algorithm R will produce, in an equivariant way, either

- a) a set A' of at most $|A|/2$ points, or
- b) a set \mathcal{P} of at most $|A|/200$ great circles.

- R1.** [Make D undirected.] Construct the undirected version of D and call it G .
- R2.** [Check for eccentric centers of mass.] Compute the center of mass of each connected component of G . If these centers are not in the origin, return the set A' of these centers.
- R3.** [Two-dimensional components?] If each component is a regular polygon with center at the origin, return the set \mathcal{P} of circumcircles of each polygon.
- R4.** [Three-dimensional components?] If each component spans a 3-dimensional hyperplane H , replace the component by two antipodal points perpendicular to H . Return the set A' of these points.
- R5.** [Toroidal grid.] As it was shown in Lemma 2.14, now each component of D is the product $P \times Q$ of a regular p -gon P and a regular q -gon Q in orthogonal planes, with $p, q \geq 3$. Pick a vertex u from each component. There are four incident edges, and the plane spanned by two incident edges is orthogonal to the plane spanned by the other two edges. Represent the component of G by these two orthogonal planes shifted to the origin. Return the set \mathcal{P} of great circles in these planes (two per component).

The algorithm takes linear time. We still need to show that these cases are exhaustive. After we make D undirected in Step R1, for every edge uv of G , the bisecting hyperplane of uv acts as a mirror that exchanges u and v together with their neighborhoods. Since the reflected mirror hyperplanes act again as mirrors, it follows that every component of the graph is the orbit of some point u_0 under the group generated by reflections perpendicular to the edges incident to u_0 . Thus, the incident edges of a single point determine the geometry of the whole component. The graph may have several components, all of which are congruent.

There are infinitely-many groups of the form $D_2^p \times D_2^q$, where D_2^p is the dihedral group of order $2p$, the symmetry group of the regular p -gon. These groups are treated in Step R5. Except this infinite family, there are only eight other groups, which can be excluded by Lemma 2.14. The details have already been described in Section 2.5.

We go through the steps one by one to check if the algorithm achieves the claimed results. Step R2 treats the case when a (lower-dimensional) component does not go through the origin. This includes the cases when G is a matching or a union of “small” regular polygons. Every component contains at least 2 points, and thus $|A'| \leq |A|/2$. Steps R3 and R4 treat the two- and three-dimensional components that are centered at the origin. (The one-dimensional case of a matching cannot be centered at the origin, because then we would have $\delta = 2$.) Suppose we have a two-dimensional component and this component is a regular k -gon inscribed in a great circle (Step R3). From $\delta < \delta_0$ and by Corollary 5.3, we obtain that $k > 12000$. Thus $|C| \leq |A|/12000 \leq |A|/200$. A three-dimensional component (Step R4) contains at least four points and is reduced to two antipodal points. Again we have $|A'| \leq |A|/2$.

Steps R2–R4 have treated all cases of Coxeter groups which are not full-dimensional, and Lemma 2.14 excludes all full-dimensional groups which are not of the form $D_2^p \times D_2^q$. Thus, in Step R5, each component of D is the product $P \times Q$ of a regular p -gon P and a regular q -gon Q in orthogonal planes, with $p, q \geq 3$.

Note that a regular p -gon can be generated as the orbit of a point u_0 in D_2^p or in D_2^{2p} , depending on whether we put u_0 in the interior of the fundamental region or on a mirror. But this makes no difference for the resulting point set.

Such a component forms a toroidal $p \times q$ grid. Each vertex has four neighbors. The two polygons P and Q have equal side lengths δ , because otherwise the four neighbors would not be part of the closest-pair graph G . The two incident edges of a vertex u that come from P are orthogonal to the two edges that come from Q , so we can distinguish the two edge classes. (The case when all four edges are perpendicular is the 4-cube with $p = q = 4$ and with reflection group $D_2^4 \times D_2^4$. This case has 16 vertices $(\pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2)$ and minimum distance $\delta = 1$ and is therefore excluded.) Moreover, the grid cells are geometric squares. Through these squares, the classification of edges into the edges from P and the edges from Q can be transferred consistently to a whole connected component of G . All copies of P in the grid lie in parallel planes, and so do the copies of Q . Accordingly, Step R5 represents each connected component by two orthogonal planes through the origin. We need to show that the component is large. As $P \times Q$ lies on the unit 3-sphere, the circumradii r_P and r_Q of the two polygons satisfy $r_P^2 + r_Q^2 = 1$. Thus, the larger circumradius, let us say r_P , is at least $1/\sqrt{2}$. Since the closest-pair distance is $\delta = 2r_P \sin \frac{\pi}{p} \leq \delta_0 = 0.0005$, we get $\sqrt{2} \sin \frac{\pi}{p} \leq \delta_0$, which implies $p \geq 8886$. Each component contains $pq \geq p$ points and is reduced to two circles. Thus, the algorithm achieves the claimed reduction.

5.4 Marking and Condensing of Great Circles: Algorithm M

We have extracted a set \mathcal{P} of at most $|A|/200$ great circles from the point set A , either from the mirror case (Algorithm R in Section 5.3) or from orbit cycles (Algorithm O in Section 5.2). Algorithm M obtains a small set A' of *marker points* on each circle so that we can resume Algorithm C, or it exits to Algorithm T for 2+2 dimension reduction (Section 5.5).

For this purpose, we look at the angles α, β of pairs of circles $C, D \in \mathcal{P}$. If C and D are not Clifford parallel, we can mark a pair of points on C and on D respectively. See before in Section 2.2 to find out angles, Clifford parallelism and marking. Doing this for all pairs of circles would lead to a quadratic blowup. Therefore we construct the closest-pair graph on the *set of circles*. We use Plücker coordinates to map great circles of \mathbb{S}^3 to points on \mathbb{S}^5 . We can then compute the closest-pair graph in six dimensions, and every circle has at most K_5 closest neighbors.

Plücker coordinates and Plücker distance were explained before in Section 2.3. Lemma 2.3 and Corollary 2.4 in Section 2.3 showed that Plücker distance is geometrically meaningful and does not depend on the choice of a coordinate system.

The marking approach fails for Clifford-parallel circles. Clifford-parallel pairs manifest a very rigid structure on a sphere. Section 2.4 described beautiful properties of such pairs. For convenience, we restate the results in Section 2.4 using the following proposition. They are formulated for right pairs of circles, but they hold equally with left and right exchanged.

- Proposition 5.4.**
1. *The relation of being a right pair is transitive (as well as reflexive and symmetric) (Corollary 2.10). 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 (Lemma 2.7).*
 3. *By this map, two Clifford-parallel circles with angle α, α (and with Euclidean distance $\sqrt{2} \sin \alpha$ on the Plücker sphere \mathbb{S}^5) are mapped to points at angular distance 2α on \mathbb{S}^2 (Lemma 2.11).*
 4. *A circle can have at most $K_2 = 5$ closest neighbors on the Plücker sphere \mathbb{S}^5 that form right pairs.*

Proof of Property 4. Property 4 is a direct consequence of Properties 2 and 3. □

We need one more ingredient before we can state the algorithm. We have seen that an auxiliary Algorithm K in Section 3.3.3. Algorithm K condenses a point set on a 2-sphere to a smaller point set of at most 12 points.

Algorithm M (*Mark and condense great circles*). Given a set \mathcal{P} of great circles on \mathbb{S}^3 , this algorithm will produce, in an equivariant way, either a nonempty set A' of at most $100|\mathcal{P}|$ points on \mathbb{S}^3 , or a nonempty set \mathcal{P}' of at most 829 great circles.

The algorithm will update the set \mathcal{P} and maintain an equivalence relation \sim on \mathcal{P} such that all circles in the same equivalence class belong to a common (left or right) Hopf bundle. The common chirality of all bundles is indicated by a variable $chirality \in \{None, Left, Right\}$, where *None* is chosen when the equivalence relation is trivial and all classes are singletons. The size of the equivalence classes is bounded by 12.

M1. [Initialize.] Let every circle $C \in \mathcal{P}$ form a singleton equivalence class.

M2. [Prune by size.] If equivalence classes are of different sizes, choose the size that occurs least frequently. Throw away all equivalence classes that are not of this size, together with the circles they contain.

- M3.** [Few circles?] If $|\mathcal{P}| \leq 829$, return the set \mathcal{P} .
- M4.** [Singletons?] If all equivalence classes are singletons, set $chirality := None$.
- M5.** [Construct closest pairs.] Represent each circle $C \in \mathcal{P}$ by two antipodal points on \mathbb{S}^5 , using Plücker coordinates. Construct the closest-pair graph H on \mathcal{P} with respect to the distances on \mathbb{S}^5 .
- M6.** [Classify edges.] Partition the edges of H into $E_L \cup E_R \cup E_N$, representing pairs of circles that are left pairs, right pairs, and not Clifford parallel.
- M7.** [Use non-Clifford-parallel edges.] If $E_N \neq \emptyset$, let $\mathcal{N} := E_N$ and go to Step M12.
- M8.** [Find non-Clifford-parallel pairs from equivalent circles.] If $E_L \neq \emptyset$ and $chirality = Right$, set $\mathcal{N}_C := \{\{C', D\} \mid \{C, D\} \in E_L, C' \text{ is closest to } C \text{ among the circles } C' \parallel_+ C, C' \neq C\}$ for all $C \in \mathcal{P}$, $\mathcal{N} := \bigcup_{C \in \mathcal{P}} \mathcal{N}_C$ and go to Step M12.
- M9.** (Same as M8, with left and right exchanged.)
- M10.** [Merge classes.] If $E_L \neq \emptyset$ and $chirality \in \{Left, None\}$, merge equivalence classes that contain circles that are connected in E_L . Use Algorithm K to condense each resulting class F to a set F' of at most 12 *representative* circles. Set \mathcal{P} to the set of all representative circles, and put them into the same equivalence class if and only if they come from the same set F' . Set $chirality := Left$, and return to M2.
- M11.** (Same as M10, with left and right exchanged. Since all possibilities are exhausted, the only remaining possibility in this step is to merge classes and return to M2.)
- M12.** [Mark points on circles.] (Each pair in \mathcal{N} is now a non-Clifford-parallel pair of circles.) For each pair of circles $\{C, D\} \in \mathcal{N}$, mark the two points on C that are closest to D , and likewise, mark two points on D . Return the set A' of all marked points.

The crucial properties for the correctness and the running time are formulated in a lemma:

- Lemma 5.5.** 1. *After the algorithm returns to Step M2 from step M10 or M11, the number of equivalence classes is reduced at least by half.*
2. *Algorithm M terminates in $O(|\mathcal{P}| \log |\mathcal{P}|)$ time.*
3. *In Step M12, \mathcal{N} is a nonempty set of at most $25|\mathcal{P}|$ non-Clifford-parallel pairs.*

Proof. 1. The only possibility for the algorithm to stall is that the edges of H are *within* classes, and no merging takes place in Step M10 or M11. Each class must be one of the five configurations listed in Lemma 3.2, and the smallest possible angular distance $2\alpha_{\min} \approx 1.10715$ occurs for two adjacent vertices of the icosahedron, and a volume packing argument on \mathbb{S}^5 yields that there can be at most 829 circles with this distance. See Section 3.1.2. Then the algorithm exits in Step M3.

2. The most expensive step is the construction of the closest-pair graph in Step M5, which takes $O(|\mathcal{P}| \log |\mathcal{P}|)$ time. The algorithm contains one loop, when returning from M10 or M11 to M2. Since the number of equivalence classes is geometrically decreasing and each class contains at most 12 circles, the overall time is also bounded by $O(|\mathcal{P}| \log |\mathcal{P}|)$.

3. When \mathcal{N} is constructed in Step M7, there can be at most $22|\mathcal{P}|$ such pairs, because the degree in H is bounded by $K_5 \leq 44$. In Step M8, the constructed set \mathcal{N} is nonempty: Every pair $\{C, D\} \in E_L$ produces at least one element of \mathcal{N} , since all equivalence classes contain at least

two elements, by M2 and M4. When a pair $\{C', D\}$ in \mathcal{N} is formed, we have a left pair $\{C, D\}$ and a right pair $\{C, C'\}$. If $\{C', D\}$ were also Clifford parallel, we would get a contradiction to transitivity (Proposition 5.4.1). Each circle $C \in \mathcal{P}$ gives rise to at most $5 \cdot 5 = 25$ pairs, since a circle can have at most 5 Clifford-parallel neighbors in H by Proposition 5.4.4. \square

We conclude that the algorithm produces at most $100|\mathcal{P}|$ points, four points for each pair in \mathcal{N} .

5.5 2+2 Dimension Reduction: Algorithm T

If we arrive at Algorithm T, we are in the very last step so we should restore the initial input point sets A and B . This 2+2 dimension reduction is applied when we have already identified at most C_1 pairs of planes P and Q from Algorithm M in Section 5.4 such that any candidates R of congruence mapping from A to B has to map P to Q .

We begin by choosing a coordinate system (x_1, y_1, x_2, y_2) for A so that P becomes the x_1y_1 -plane, and similarly for B and Q . We then look for rotations R that leave the x_1y_1 -plane invariant. Such rotations have the form $R = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}$ with two orthogonal 2×2 matrices R_1 and R_2 . Since $\det R = \det R_1 \cdot \det R_2 = 1$, we try two cases: (a) R_1 and R_2 are planar rotations; (b) R_1 and R_2 are planar reflections. We can reduce (b) to (a) by applying the rotation $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ to A . Thus, it suffices to describe (a), where R has the form $R_{\varphi, \psi}$ in (2.1), i. e., a combination of a rotation by φ in the x_1y_1 -plane and an independent rotation by ψ in the x_2y_2 -plane. We use polar coordinates in these two planes by setting

$$\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} r_1 \cos \alpha_1 \\ r_1 \sin \alpha_1 \\ r_2 \cos \alpha_2 \\ r_2 \sin \alpha_2 \end{pmatrix} \quad \text{with } r_1 = \sqrt{x^2 + y^2} \text{ and } r_2 = \sqrt{z^2 + w^2}. \quad (5.5)$$

The rotation $R_{\varphi, \psi}$ changes (α_1, α_2) by adding (φ, ψ) modulo 2π , and leaves r_1 and r_2 unchanged. In other words, $R_{\varphi, \psi}$ acts as a translation on the torus $\mathbb{T}^2 = [0, 2\pi) \times [0, 2\pi)$. We attach the distances (r_1, r_2) to each point $(\alpha_1, \alpha_2) \in \mathbb{T}^2$ as a *label*. The problem becomes therefore a *two-dimensional problem* of testing *congruence under translation* for *labeled* points on a torus. We denote the two labeled point sets as \tilde{A} and \tilde{B} , and a point of \tilde{A} can only be mapped to a point of \tilde{B} with the same label.

Points in the x_1y_1 -plane should be considered separately, because α_2 is not unique when $r_2 = 0$, and the same problem occurs for the points in the x_2y_2 -plane. We will defer the treatment of these points to the end of this section, and start with other points first.

We now give an algorithm for the following problem: given two labeled point sets \tilde{A} and \tilde{B} on the torus \mathbb{T}^2 , test if \tilde{A} and \tilde{B} are the same up to translations. We will find a canonical set of \tilde{A} (and \tilde{B}) which is similar to a condensed set. In contrast to a condensed set, we add no new symmetries to a canonical set, by updating labels to preserve complete information. Let $\text{Sym}(A)$ for a set $A \subset \mathbb{T}^2$ denote translational symmetry group of A , i. e., the set of translations that map A to itself and preserve labels, if A is a labeled set.

5.5.1 A Canonical Set Procedure

Roughly, our goal is to find simplest subsets A' and B' of representative points that still preserve the same symmetries as A and B respectively. In addition, we want to construct A' by no arbitrary decisions. If we can find such subsets, it is enough to compare an arbitrary point in A' with that in B' . We construct A' for a given point set A in some space \mathcal{S} and some group Θ of symmetries of \mathcal{S} in two cases: (i) $\mathcal{S} = \mathbb{S}^1$ (the unit circle), and Θ are the rotations of \mathbb{S}^1 . (ii) $\mathcal{S} = \mathbb{S}^1 \times \mathbb{S}^1$ (the flat torus), and Θ are the translations on \mathcal{S} .

Now, we formally define a *canonical set procedure* and explain how to use it. We denote by $\text{Sym}_{\Theta}(A)$ the symmetry group of A within Θ :

$$\text{Sym}_{\Theta}(A) = \{ R \in \Theta \mid R(A) = A \text{ where } R \text{ preserves labels} \}.$$

Definition 5.6. A *canonical set procedure* f_{cano} for a space \mathcal{S} and a subgroup Θ of the symmetries of \mathcal{S} maps every finite set $A \subset \mathcal{S}$ to a set $A' \subset \mathcal{S}$ such that the following properties hold.

1. Symmetries are preserved: $\text{Sym}_{\Theta}(A') = \text{Sym}_{\Theta}(A)$
2. $\text{Sym}_{\Theta}(A')$ acts transitively on A' : For every $p, q \in A'$, there exists $R \in \text{Sym}_{\Theta}(A')$ that maps p to q .
3. A' is defined in an equivariant way from A : If $RA = B$ for some $R \in \Theta$, and B is mapped to B' , then $B' = RA'$.

We call A' a *canonical set* of A .

It follows from the definition that the canonical set A' is nonempty whenever A is nonempty, provided that Θ is an infinite group. It is easy to see that condition 3 is implied by the \supseteq part of condition 1. Thus, if the procedure is equivariant, we only have to prove that A' does not have more symmetries than A .

A canonical set procedure is used to check congruence of two sets $A, B \subset \mathcal{S}$ under a congruence from the class Θ as follows: We compute their canonical point sets A' and B' . We then pick two arbitrary points $p \in A'$ and $q \in B'$, we find the unique rotation $R \in \Theta$ that maps p to q , and finally, we simply have to check whether $RA = B$. In the above two cases of \mathcal{S} and Θ , there is always such a unique R .

The correctness of this approach is formulated in the following easy lemma.

Lemma 5.7. *Suppose that \mathcal{S} is a space in which, for any two points $p, q \in \mathcal{S}$, there is a unique symmetry $R = R_{pq} \in \Theta$ with $Rp = q$. Let A' and B' be canonical sets of A and B for \mathcal{S} and Θ respectively. There exists a congruence in Θ that maps A to B if and only if for any $p \in A', q \in B'$, we have $R_{pq}A = B$.*

Proof. Suppose there is a congruence $R_0 \in \Theta$ such that $R_0(A) = B$ and $p \in A'$ and $q \in B'$ are chosen arbitrarily. Since A' and B' are obtained by a canonical set procedure, $B = R_0(A)$ is mapped to $B' = R_0(A')$. Let p' be the pre-image of q under R_0 . Since A' is a canonical point set, there exists a unique $R_1 \in \text{Sym}_{\Theta} A' \subset \text{Sym}_{\Theta} A$ satisfying $R_1(p) = p'$ and $R_1(A) = A$. By taking $R = R_0 \cdot R_1$, R is the unique rotation satisfying $R(p) = R_0(R_1(p)) = R_0(p') = q$ and $R(A) = R_0(R_1(A)) = R_0(A) = B$. Also, $R \in \Theta$ since $R_0, R_1 \in \Theta$ and Θ is closed under the multiplication.

The other direction is obvious. □

In the case (i) that $\mathcal{S} = \mathbb{S}^1$, the map from a set of points in \mathbb{S}^1 to a set of *canonical axes* (refer to Section 3.2.1) is the obvious canonical set procedure. It yields a set of regularly spaced directions on \mathbb{S}^1 .

A canonical set procedure for case (ii) is presented in Lemma 5.8 with Algorithm T.

Algorithm T (*A canonical set procedure from a labeled point set on the torus*). The input is a labeled point set \tilde{A} on the torus \mathbb{T}^2 . We assume that two labels can be compared in constant time. The output is an unlabeled *canonical set* \hat{A} as in Definition 5.6. In addition, \hat{A} should be obtained from \tilde{A} without making any arbitrary decisions.

- T1.** [Prune by labels.] Choose the label that occurs least frequently in \tilde{A} , and let A' be the set of points with this label. (For a while we will now do ordinary pruning, using only the geometry of the point set A' .)
- T2.** [Compute the Voronoi diagram.] Compute the Voronoi diagram of A' on the torus \mathbb{T}^2 . This can be reduced to a Voronoi diagram in the plane by replicating the square region

representing the torus together with the set A' 9 times in a 3×3 tiling, see for example [17]. Clipping the result to the central tile yields the Voronoi diagram on the torus in $O(|A'| \log |A'|)$ total time.

- T3.** [Prune by shape.] Translate each point $a \in A'$ to the origin together with its Voronoi cell. If the translated cells are not all equal, replace A' by the subset of points whose cell shape occurs least frequently, and return to T2.
- T4.** [Restore information from the original set \tilde{A} by labeling the points in A' .] (Now all Voronoi cells are translated copies of the same hexagon or rectangle.) Assign each point of \tilde{A} to its Voronoi cell. A point on the boundary is assigned to all incident cells. Now for each point $x \in A'$, collect the points in its cell and translate them so that x lies at the origin. Represent each point as a triple (φ -coordinate, ψ -coordinate, label). Concatenate these triples in lexicographic order into a string of numbers that represents the cell contents, and attach the string as a label to the point x . (This string representation is obtained equivariantly, since two points $x, y \in A'$ get the same string if and only if the two Voronoi cells are exact translated copies of each other, including all points in the cells with their original labels. We have thus preserved complete information about the symmetries of \tilde{A} .)
- T5.** [Compress labels.] Sort the strings and replace each label with its rank in sorted order.
- T6.** [Finalize.] If there are at least two labels, return to T1. Otherwise, return A' as the canonical set \hat{A} .

Lemma 5.8. *Algorithm T computes a canonical set \hat{A} of a labeled set \tilde{A} on the torus in time $O(|\tilde{A}| \log |\tilde{A}|)$.*

Proof. We first check that \hat{A} has the claimed properties. Property 5.6.1 consists of two inclusions: We have $\text{Sym}(\tilde{A}) \subseteq \text{Sym}(\hat{A})$, because \hat{A} is obtained in a equivariant way from \tilde{A} . None of the operations which are applied to \tilde{A} to obtain \hat{A} destroys any translational symmetries. The other inclusion $\text{Sym}(\hat{A}) \subseteq \text{Sym}(\tilde{A})$ is ensured by Step T4. (This would work for any set A' .)

To see Property 5.6.2, note that the Voronoi cell of a point fixes the relative positions of its neighbors. Starting from any point $a \in \hat{A}$ we can reconstruct the whole set \hat{A} if we know the shape of each Voronoi cell. Step T3 ensures that all Voronoi cells are equal, and therefore the reconstructed set $\hat{A} - a$ is the same, no matter from which point a we start. The set \hat{A} forms a lattice on the torus.

Let us analyze the running time. Each iteration of the loop T2–T3 takes $O(|A'| \log |A'|)$ time and reduces the size of A' to half or less. Thus, the total running time of this loop is $O(|\tilde{A}| \log |\tilde{A}|)$, since the initial size of A' is bounded by the size of \tilde{A} .

Steps T4 and T5 involve point location in Voronoi diagrams and sorting of strings of numbers of total length $O(|\tilde{A}|)$. These operations can be carried out in $O(|\tilde{A}| \log |\tilde{A}|)$ time. Thus, each iteration of the whole loop T1–T6 takes $O(|\tilde{A}| \log |\tilde{A}|)$ time. Step T1 reduces the size of \tilde{A} to a half or less after each iteration. Thus, the total running time is $O(|\tilde{A}| \log |\tilde{A}|)$. \square

We are ready for the main result of this section.

Theorem 5.9. *Given two sets A, B of n points, and two planes P and Q , it can be checked in $O(n \log n)$ time if there is a congruence that maps A to B , and P to Q .*

Proof. We have already seen how to convert input points to points $(\alpha_1, \alpha_2) \in \mathbb{T}^2$ with labels (d_1, d_2) so that congruence testing is reduced to testing congruence by translation on the torus.

We still have to deal with points on coordinate planes. Let $A_1 \subseteq A$ be the points in the x_1y_1 -plane, and let $A_2 \subseteq A$ be the points in the x_2y_2 -plane. If $A_1 \neq \emptyset$, we compute the canonical axes of A_1 , as described in Section 3.2.1. They form $k \geq 1$ equally spaced angular directions $\bar{\alpha} + j\frac{2\pi}{k}$ modulo 2π , $j \in \mathbb{Z}$. We know that R must map the canonical axes of A_1 to the canonical axes for the corresponding set B_1 . We incorporate this restriction into an additional label for each point $u \in A \setminus A_1 \setminus A_2$. If u has a polar angle α_1 in the x_1y_1 -plane, we attach to it the difference to the nearest smaller canonical angle:

$$\min\{\alpha_1 - (\bar{\alpha} + j\frac{2\pi}{k}) \mid j \in \mathbb{Z}, \alpha_1 - (\bar{\alpha} + j\frac{2\pi}{k}) \geq 0\}$$

We also have to test if A_1 and B_1 are congruent by simply overlaying the canonical axes and testing if they are equal.

If $A_2 \neq \emptyset$, we treat this in the same way and attach an additional label to the points in $A \setminus A_1 \setminus A_2$.

After this, we can just compute the canonical set from Lemma 5.8 and thereby reduce the congruence test to an equality test between sets \hat{A} and \hat{B} , as described before in Lemma 5.7. If there are no points outside the xy -plane and the zw -plane, the problem reduces to two independent congruence tests in these planes. The whole algorithm takes $O(n \log n)$ time. \square

There is an alternative algorithm with the same time complexity that does not use Lemma 5.7.

Alternative Proof of Theorem 5.9. We can also use Algorithm T for comparing two labeled sets \tilde{A} and \tilde{B} on the torus as follows. We run Algorithm T in parallel for \tilde{A} and \tilde{B} . We must make sure that all steps run identically. In particular, the sorted strings in Step T5 must be the same for \tilde{A} and \tilde{B} . If the algorithm runs to the end without finding a difference between \tilde{A} and \tilde{B} , and if the Voronoi cells of the canonical sets \hat{A} and \hat{B} are equal, we know that \tilde{A} and \tilde{B} are congruent by translation. (There are $|\hat{A}| = |\hat{B}|$ translations that map \hat{A} to \hat{B} .) Points on coordinate planes can be taken care of by the same way as the previous proof and the overall algorithm also takes $O(n \log n)$ time. \square

III

More Implications

This part discusses the generalization of the algorithm to five or higher dimensions and the extendability of the algorithm in other directions. This part contains speculations about possible further development so it is less concrete and rigorous. Still, it is hoped that these directions will provide helpful insights to future researchers.

Further Directions to Higher-dimensional Space

Generally, the congruence testing problem in d -space with no restriction on $d > 0$ is equivalent to the graph isomorphism problem, which determines whether two finite graphs are isomorphic or not by the following argument: any given graph $G = (V, E)$, $V = \{v_1, v_2, \dots, v_n\}$ can be represented by $n + |E|$ points in n -space by replacing each edge $v_i v_j \in E$ with a point located at $(e_i + e_j)/2$ where e_i and e_j are i th and j th standard vectors. Then two graphs are isomorphic if and only if their corresponding representations are congruent.

This chapter explores the possibility to find a fixed-parameter-tractable algorithm with the dimension parameter d . In other words, we want to know, in d -space with a fixed integer d , if there exists an algorithm with running time $O(f(d)n^{O(1)})$ where $f(d)$ is an arbitrary function depending only on d .

The new algorithm in Part II is specific to 4-space and cannot be extended to d -space for $d > 4$. The main reason is that Algorithm M in Section 5.4 uses the Hopf fibration from a 3-sphere to a 2-sphere.

The running time of the algorithms for general dimensions can be improved by incorporating our four-dimensional algorithm when the dimension is reduced to 4. The deterministic algorithm by Brass and Knauer can run in $O(n^{\lceil (d-1)/3 \rceil} \log n)$ time, and the randomized algorithm by Akutsu can perform in time $O(n^{\lfloor (d-1)/2 \rfloor / 2} \log n)$ for $d \geq 9$ and $O(n^{3/2} \log n)$ for $d \leq 8$ by starting from our 4-dimensional algorithm as the base case.

The approaches here are not yet complete but try to develop ideas for higher dimensions.

6.1 In 5 Dimensions

Although the algorithm in Part II cannot be generalized, some other modules are likely to be extended. In particular, the modules for constructing orbit cycles in Section 5.1 and Section 5.2 are probably extensible by considering a 5-tuple of points that forms a path in a closest-pair graph instead of looking at a quadruple. Provided that orbit cycles can be constructed for a point set in 5-space by extending these modules, congruence testing in 5-space can be done by simply tweaking the 4-dimensional algorithm in Part II.

Using the Plücker embedding, we can construct the closest-pair graph for a set of great circles corresponding to computed orbit cycles. By the pruning principle, we may assume that every pair of great circles has the same angle (the notion of angles between planes does not depend on the dimension of ambient space). If a pair P, Q of great circles for each closest pair spans a 3-dimensional subspace, we replace P, Q with a pair of points, which is the intersection of P and Q . Otherwise, we replace P, Q with a pair of antipodal points, whose direction is orthogonal to the space that is spanned by P, Q . If these replacements reduce the number of points to less than a half, this works as a successful condensing step. Otherwise, each orbit cycle must

be short: we can apply the same packing arguments as before to show that there are only a bounded number of points.

6.2 In High Dimensions

In fact, Algorithm T in Section 5.5 is an extension of the canonical axes in the one-dimensional torus \mathbb{S}^1 to the canonical set in the two-dimensional torus \mathbb{T}^2 . An analogue of Theorem 5.9 in Section 5.5 in higher-dimensional space is as follows: for even $d > 0$, if we have identified a pair of $d/2$ -tuples $\langle P_1, \dots, P_{d/2} \rangle$ and $\langle Q_1, \dots, Q_{d/2} \rangle$ of orthogonal planes, it can be tested whether there exists a congruence mapping that maps A to B and P_i to Q_i for $1 \leq i \leq d/2$.

Similarly to Section 5.5, we can choose the coordinate system for A and B so that those $d/2$ -tuples of orthogonal planes become parallel to coordinate planes and we can also endow polar coordinates by extending (5.5) as follow. A coordinate of a point p can be represented by

$$\begin{pmatrix} r_1 \cos \alpha_1 \\ r_1 \sin \alpha_1 \\ \dots \\ r_i \cos \alpha_i \\ r_i \sin \alpha_i \\ \dots \\ r_{d/2} \cos \alpha_{d/2} \\ r_{d/2} \sin \alpha_{d/2} \end{pmatrix} \quad (6.1)$$

where $(r_i \cos \alpha_i, r_i \sin \alpha_i)$ is the polar coordinate of the projection of p to the plane P_i (or respectively Q_i) for $i = 1, \dots, d/2$. Therefore the point p can be regarded as a point $(\alpha_1, \dots, \alpha_{d/2})$ in a $(d/2)$ -torus with a label $(r_1, \dots, r_{d/2})$.

Now, the problem is again reduced to testing translations for labeled points on a t -torus where $t = \lfloor d/2 \rfloor$. This problem can be solved in $O(n^2 \log n)$ time by the following lemma. This lemma is not a direct extension of Section 5.5 because the following algorithm does not use a canonical set procedure.

Lemma 6.1. *The algorithm D tests if two labeled sets \hat{A} and \hat{B} of n points on a t -torus are the same up to translations on the torus in time $O(n^2)$ for any $t > 0$.*

Algorithm D (*Test translations of labeled sets on a high-dimensional torus*)

- D1.** [Labeling.] Pick an arbitrary point $a \in \hat{A}$. Let s_a be the lexicographically-sorted sequence of pairs $(a_i - a, L_i)$ for all $a_i \in \hat{A}$ such that $a_i \neq a$ where L_i is the label of a_i . This sequence is of length $O(n)$. Here, each coordinate is represented in polar coordinates as in (5.5). Add s_a to the label of a .
- D2.** [Finding a matching.] For all $b \in \hat{B}$, add similar labels s_b to the label of b and find if there exists $b \in \hat{B}$ whose label is the same as that of a . If such a point b exists, \hat{A} and \hat{B} are the same up to translations on the torus and otherwise not.

Proof. If there is a translation that maps \hat{A} to \hat{B} while it preserves labels, this translation needs to map some point $a \in \hat{A}$ to some point $b \in \hat{B}$ of the same label. If there is such a pair of points a, b , the translation from a to b maps \hat{A} to \hat{B} . Therefore, \hat{A} and \hat{B} are the same up to translations on a t -torus if and only if there exists a pair of points a, b of the same label $s_a = s_b$. A pair of sorted strings of length n can be compared in $O(n)$ time. These labels are of length n , and once one string is sorted in time $O(n \log n)$, all the other strings can be computed in time $O(n)$ by additions. Thus, the total running time of Algorithm D is $O(n^2 + n \log n) = O(n^2)$. \square

We also combine the observations from Section 6.1. Suppose that we can find a set of $O(1)$ $\lfloor d/2 \rfloor$ -tuples of orbit cycles on a d -sphere from each given set and the great circles corresponding to those orbit cycles span d - or $(d - 1)$ -space. Then, we can decide if two n -point sets are congruent in d -space for any fixed $d \geq 6$ in $O(n^2)$ time.

If each tuple of great circles spans d -space, by using Gram-Schmidt orthogonalization process, there are only $(d/2)!$ ways to represent d -space by a $(d/2)$ -torus. By trying all $(d/2)!$ ways and applying Lemma 6.1, congruence of two sets can be tested in $O(d!n^2) = O(n^2)$ time if $d > 0$ is fixed.

If each tuple of great circles spans $(d - 1)$ -space, we can replace each $\lfloor d/2 \rfloor$ -tuple with a pair of antipodal points orthogonal to that $(d - 1)$ -space. If this does not work as a successful condensing, then we know the number of points is bounded by the packing argument.

The possibility that $\lfloor d/2 \rfloor$ -tuples degenerate to lower-dimensional space than $(d - 1)$ -space can be excluded by the following induction argument:

The base case is similar to the argument in Section 6.1. We can mark the intersection of a pair of great circles spanning 3-space. Then, this is a successful condensing and otherwise we can apply the packing argument. Thus, we may assume every pair of great circles span 4-space.

Suppose all (ordered) k -tuples of great circles span a $2k$ -dimensional subspace ($2k \leq d$). If a $(k + 1)$ -tuple spans a $(2k + 1)$ -subspace, we can replace this $(k + 1)$ -tuple with a pair of antipodal points in the $(2k + 1)$ -subspace orthogonal to the $2k$ -subspace spanned by the first k -tuple. This should work as a successful condensing and otherwise we should be able to apply the similar packing argument. Hence, we may assume every $(k + 1)$ -tuple spans $(2k + 2)$ -subspace.

However, this overall plan is still incomplete due to the possibility that no $\lfloor d/2 \rfloor$ -tuple is found in the closest-pair graph of great circles. For example, in 12-space, each connected component of the closest-pair graphs of great circles might be of size 3, spanning 6-subspace.

Concluding Remarks

7.1 Implementability

Instead of using the Real-RAM model, it also makes sense to test congruence with an error tolerance ε , but this problem is known to be NP-hard even in the plane [16, 24] as mentioned earlier in Section 1.2. However, the problem becomes polynomial if the input points are sufficiently separated compared to ε . We are confident that our algorithm, when it is implemented with standard floating-point arithmetic and with appropriate error margins to shield equality tests from round-off errors, would decide approximate congruence in the following weak sense. Given a tolerance ε that is large compared to the machine precision, and two sets A and B whose points are separated by more than, say 10ε , the algorithm would correctly distinguish the instances that are congruent with a tolerance of ε from the instances that are not even congruent with a tolerance of, say 100ε . Between ε and 100ε , the algorithm is allowed to make errors. Such a result will require a thorough analysis of the numerical stability of the operations in our algorithm.

7.2 Regularity and Related Open Questions

Local regularity implies global regularity? The general theme in our four-dimensional algorithms arises the question whether *local* regularity implies *global* regularity. In other words, when the neighborhoods of all the points look the same, does it imply that a symmetry group acts transitively on the point set or not, if the neighborhoods are sufficiently large? It would be interesting to prove this statement quantitatively.

Geometric classification of 4-dimensional point groups. The four-dimensional point groups were already enumerated [35, 13]. The question is whether the four-dimensional point group can be enumerated with geometrically interpretable characteristics, as it has done for the three-dimensional point group by Hessel's Theorem [30]. The following is a conjecture by Rote [33].

Conjecture 7.1. *A four-dimensional point group is either*

- (a) *the symmetry group of one of the six four-dimensional regular polytope;*
- (b) *a direct product of lower-dimensional points groups; or*
- (c) *a subgroup of one of the above groups.*

Tilings on a sphere. Another interesting question is the classification of regular and semiregular tilings on spheres. It is possible that our techniques can shed light on symmetries on the 3-sphere, or on regular and semiregular tilings of \mathbb{S}^3 .

7.3 Summary and Future Work

This thesis provided an optimal congruence testing algorithm in 4-space and showed an attempt to generalize this method to higher dimensions. Hopefully, the algorithm together with the survey in this thesis will help to settle the congruence testing problem in any fixed dimension.

Bibliography

- [1] T. Akutsu. On determining the congruence of point sets in d dimensions. *Computational Geometry: Theory and Applications*, 4(9):247–256, 1998.
- [2] H. Alt, K. Mehlhorn, H. Wagnen, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete & Computational Geometry*, 3(1):237–256, 1988.
- [3] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing*, 6(1):128–143, 1985.
- [4] M. J. Atallah. On symmetry detection. *IEEE Transactions on Computers*, 100(7):663–666, 1985.
- [5] M. D. Atkinson. An optimal algorithm for geometrical congruence. *Journal of Algorithms*, 8(2):159–172, 1987.
- [6] J. L. Bentley and M. I. Shamos. Divide-and-conquer in multidimensional space. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 220–230. ACM, 1976.
- [7] M. Berger. *Geometry, Volume II*. Springer-Verlag, New York, 1996.
- [8] M. Berger. *Geometry I*. Springer Science & Business Media, 2009.
- [9] Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.
- [10] P. Braß and C. Knauer. Testing the congruence of d -dimensional point sets. *International Journal of Computational Geometry and Applications*, 12(1&2):115–124, 2002.
- [11] P. Brass and C. Knauer. Testing congruence and symmetry for general 3-dimensional objects. *Computational Geometry*, 27(1):3–11, 2004.
- [12] J. H. Conway, R. H. Hardin, and N. J. Sloane. Packing lines, planes, etc.: Packings in Grassmannian spaces. *Experimental Mathematics*, 5(2):139–159, 1996.
- [13] J. H. Conway and D. A. Smith. On quaternions and octonions: their geometry, arithmetic, and symmetry. *AK Peters, Wellesley, Massachusetts*, 2003.
- [14] H. S. M. Coxeter. *Regular Polytopes*. Dover Publications, 3rd edition, 1973.
- [15] P. J. de Rezende and D. Lee. Point set pattern matching in d -dimensions. *Algorithmica*, 13(4):387–404, 1995.
- [16] C. Dieckmann. *Approximate Symmetries of Point Patterns*. PhD thesis, Freie Universität Berlin, 2012.

- [17] N. P. Dolbilin and D. H. Huson. Periodic Delone tilings. *Periodica Mathematica Hungarica*, 34(1-2):57–64, 1997.
- [18] W. Fenchel. Über Krümmung und Windung geschlossener Raumkurven. *Mathematische Annalen*, 101(1):238–252, 1929.
- [19] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.
- [20] P. T. Highnam. Optimal algorithms for finding the symmetries of a planar point set. *Information Processing Letters*, 22(5):219–222, 1986.
- [21] J. E. Hopcroft and R. E. Tarjan. A $v \log v$ algorithm for isomorphism of triconnected planar graphs. *Journal of Computer and System Sciences*, 7(3):323–331, 1973.
- [22] H. Hopf. Über die Abbildungen der dreidimensionalen Sphäre auf die Kugelfläche. *Mathematische Annalen*, 104:637–665, 1931.
- [23] H. Hopf. Über die Abbildungen von Sphären auf Sphären niedrigerer Dimension. *Fundamenta Mathematicae*, 25(1):427–440, 1935.
- [24] S. Iwanowski. Testing approximate symmetry in the plane is NP-hard. *Theoretical Computer Science*, 80(2):227–262, 1991.
- [25] X. Jiang, K. Yu, and H. Bunke. Detection of rotational and involutorial symmetries and congruity of polyhedra. *The Visual Computer*, 12(4):193–201, 1996.
- [26] H. Kim and G. Rote. Congruence testing for point sets in 4-space. In *Proceedings of 32th Annual Symposium on Computational Geometry*, 2016, to appear.
- [27] D. E. Knuth, J. H. Morris, Jr, and V. R. Pratt. Fast pattern matching in strings. *SIAM journal on computing*, 6(2):323–350, 1977.
- [28] G. Manacher. An application of pattern matching to a problem in geometrical complexity. *Information Processing Letters*, 5(1):6–7, 1976.
- [29] H. P. Manning. *Geometry of Four Dimensions*. Macmillan, 1914.
- [30] G. E. Martin. *Transformation geometry: An introduction to symmetry*. Springer, 1982.
- [31] F. P. Preparata and M. Shamos. *Computational geometry: an introduction*. Springer Science & Business Media, 2012.
- [32] E. M. Reingold. On the optimality of some set algorithms. *Journal of the ACM*, 19(4):649–659, 1972.
- [33] G. Rote. Congruence testing of point sets in three and four dimensions, results and techniques. In *Proceedings of the Sixth Conference on Mathematical Aspects of Computer and Information Sciences, Lecture Notes in Computer Science*. Springer-Verlag, 2016, to appear.
- [34] K. Sugihara. An $n \log n$ algorithm for determining the congruity of polyhedra. *Journal of Computer and System Sciences*, 29(1):36–47, 1984.
- [35] W. Threlfall and H. Seifert. Topologische Untersuchung der Diskontinuitätsbereiche endlicher Bewegungsgruppen des dreidimensionalen sphärischen Raumes. *Mathematische Annalen*, 104(1):1–70, 1931.

- [36] J. D. Ullman, A. V. Aho, and J. E. Hopcroft. The design and analysis of computer algorithms. *Addison-Wesley, Reading*, 4, 1974.
- [37] Y.-C. Wong. Differential geometry of grassmann manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 57(3):589, 1967.

Zusammenfassung

Kongruenz ist ein Konzept der Geometrie, das die Gleichheit von Objekten in euklidischen Räumen beschreibt. Die erlaubten Operationen sind Rotationen und Translationen. Da Kongruenz eines der grundlegendsten Konzepte in der Geometrie ist, beschäftigen sich viele Wissenschaftler der algorithmischen Geometrie mit der Frage, ob der Test auf Kongruenz von zwei Punktmengen dem sogenannten „Fluch der Dimensionalität“ unterliegt. In dieser Arbeit entwickeln wir einen deterministischen Algorithmus für den Kongruenztest von Punktmengen in vierdimensionalen Räumen mit optimaler Laufzeit.

Um die Bedeutung des Hauptalgorithmus im historischen Zusammenhang zu verstehen, beginnen wir mit einem Überblick über das zu Grunde liegende Rechenmodell. Weiterhin rezensieren wir die für Kongruenztestalgorithmen relevanten bisherigen Ergebnisse.

Anschließend werden die ausschlaggebenden Bestandteile des neu entwickelten Algorithmus beschrieben. Dies beinhaltet insbesondere allgemeine vierdimensionale Drehungen, Winkel zwischen linearen Unterräumen und Plücker Einbettungen. Bei seiner Ausführung forciert der Algorithmus durch eine gezielte Fallunterscheidung in jedem Schritt mehr und mehr regelmäßige Strukturen in der Punktmenge. Dabei stoßen wir auf ansprechende mathematische Strukturen auf 3-Sphären und Spiegelungsgruppensymmetrien, die es zu verstehen und zu analysieren gilt. Dies beinhaltet die Hopf-Faserung einer dreidimensionalen Sphäre und die Coxeter Gruppe vierdimensionaler Spiegelungsgruppen. Aus diesem Grund geben wir eine einfache und geschlossene Übersicht über diese beiden Themen.

Der Hauptalgorithmus besteht aus fünf Modulen, wobei jedes für sich einen anspruchsvollen Teilalgorithmus bildet. Das führt zu einem komplexen Hauptalgorithmus und zwingt uns zu pessimistischen Abschätzungen der Konstanten in der asymptotischen Laufzeit. Trotzdem kann dieser Algorithmus als ein erster, wichtiger Schritt zur Entwicklung von effizienten Algorithmen in höheren Dimensionen angesehen werden. Wir untersuchen mögliche Ansätze für diese Erweiterungen. Zum Abschluss folgt eine Diskussion über die Implementierbarkeit und über weitere mögliche geometrische Eigenschaften von vierdimensionalen Punktmengen, die der Algorithmus ausnutzt.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich alle Hilfsmittel und Hilfen angegeben habe und versichere, auf dieser Grundlage die Arbeit selbständig verfasst zu haben. Die Arbeit habe ich nicht in einem frühen Promotionsverfahren eingereicht.

Berlin, den 8. März 2016