JOURNAL OF ALGORITHMS 8, 159–172 (1987)

An Optimal Algorithm for Geometrical Congruence

M. D. ATKINSON*

School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6

Received October 14, 1983

An algorithm is given which, in time $O(n \log n)$, determines all the Euclidean congruences (if any) between two n-point sets in 3-dimensional space. The algorithm is shown to be optimal to within a constant factor. © 1987 Academic Press, Inc.

1. INTRODUCTION

A congruence between two objects S and T in 3-dimensional space is a mapping composed of a rotation followed by a translation (followed, possibly, by a reflection) which transforms S onto T. If it exists, such a transformation demonstrates that S and T are essentially identical save for their position in space (and, possibly, their orientation).

Let $\mathrm{Cong}(S, T)$ denote the set of all congruences between S and T. A classical problem in geometry is to decide whether $\mathrm{Cong}(S, T)$ is nonempty. There may be many congruences between S and T; in fact it is easy to see that, when $\mathrm{Cong}(S, T)$ is nonempty, it is a complete coset of the subgroup $\mathrm{Cong}(S, S)$ in the group of all symmetries of 3-dimensional space. Informally, then, there can be many congruences only if S is highly symmetrical.

In this paper we shall consider the discrete case that S and T are finite n-point sets and develop an algorithm to determine all the congruences between them. The complexity of the algorithm will be shown to be $O(n \log n)$ and we shall also show that this cannot be improved. We shall assume the standard RAM model of computation and assume also that arithmetic operations are to indefinite precision. The data to the algorithm will be assumed to be the two lists of n Cartesian triples which describe the sets S and T (but, of course, the coordinatisation is not important).

*This research was partly supported by the National Science and Engineering Research Council of Canada under Grant A2419.

0196-6774/87 $3.00
Copyright © 1987 by Academic Press, Inc.
All rights of reproduction in any form reserved.

It is easy to see that, in time $O(n)$, S and T may be translated so that their centroids are at the coordinate origin. We shall assume that this has always been done, in which case the set Cong(S, T) consists of rotations about the origin followed perhaps by a reflection. In fact it is sufficient to find all rotation congruences; for, if this is possible, we can find the full set of congruences by first finding the rotations which transform S to T and then finding the rotations which transform S to a mirror image of T. Thus the central problem in computing Cong(S, T) is to compute the set Rot(S, T) of rotations about the origin which transform S to T.

If the sets S and T are point sets on a line (the 1-dimensional case) a congruence rotation exists if and only if $S = T$ or $S = -T$. Consequently determining whether Rot(S, T) is nonempty is no easier than testing for set equality; therefore, by virtue of [4], we may conclude that the time complexity of the congruence problem is $\Omega(n \log n)$. The main part of this paper is the description of an algorithm which attains this lower bound. This algorithm, together with the remarks above show that the congruence problem has time complexity $\theta(n \log n)$.

If S and T lie in a plane (the 2-dimensional case) a technique first noted by Manacher [2] can be used to derive an $O(n \log n)$ algorithm (or, assuming the data are appropriately sorted initially, a linear time algorithm). In the 3-dimensional case, for the special situation that S and T are the vertex sets of polyhedra, Sugihara [5] has given an $O(n \log n)$ congruence testing algorithm (but note that, in his work, n is the number of edges of the polyhedra).

Before giving the algorithm for determining the complete set Rot(S, T) for arbitrary sets S and T we observe that a simple algebraic calculation will solve the problem in most cases. A congruence rotation can be described in analytic terms as a transformation on 3-vectors of the form

$$\mathbf{x} \to R\mathbf{x},$$

where R is some orthogonal matrix whose existence must be determined. Let A be the $3 \times n$ matrix whose columns are the Cartesian coordinates of the points in the set S, and let B be the similarly defined matrix for the set T. If it exists, the matrix R must satisfy

$$RA = BP,$$

where P is some $n \times n$ permutation matrix (this equation states that R transforms the points of S to those of T in some order). Since $P^{-1} = P^{\mathrm{T}}$ (the transpose of P) and $R^{-1} = R^{\mathrm{T}}$ it follows that

$$RAA^{\mathrm{T}} = BB^{\mathrm{T}}R$$

and hence R maps the eigenspaces of the 3×3 matrix AA^{T} to those of

BB^{T} (with the same eigenvalue). In particular, if AA^{T} and BB^{T} do not have the same set of eigenvalues then S cannot be congruent to T. If AA^{T} and BB^{T} do have the same set of eigenvalues and these are distinct then the orthogonal matrix R is determined uniquely by its effect on the one-dimensional eigenspaces; we must then test whether R is indeed a congruence between S and T. The argument fails if the sets of eigenvalues of AA^{T} and BB^{T} are the same but not distinct. Unfortunately, this happens whenever there is more than one congruence between the point sets (and so they have nontrivial symmetries; from a geometrical point of view, this is perhaps the most interesting case).

2. The Congruence Algorithm

2.1. *Discussion*

We shall give the algorithm in a number of major steps, justify each step, and give the execution time analysis of each step. A central concept in the algorithm is the idea of a *stable pair* of sets; such a pair S^*, T^* essentially allows all the congruences between S and T to be recovered from the set $\mathrm{Rot}(S^*, T^*)$ even though S^* and T^* may be much smaller in size than S and T. Formally we define two nonempty sets S^*, T^*, neither of them containing the origin and of the same size, to be a stable pair (with respect to S and T) if every rotation which maps S onto T maps S^* onto T^*; in other words $\mathrm{Rot}(S^*, T^*) \supseteq \mathrm{Rot}(S, T)$. Notice that we do not require that equality holds. Of course, if S^* and T^* are not rotationally congruent then neither are S and T but the converse may be false. However, even if $\mathrm{Rot}(S^*, T^*)$ is nonempty it may be possible to identify the subset $\mathrm{Rot}(S, T)$ efficiently.

The algorithm falls into two separate parts. In the first part a stable pair S^*, T^* is found with S^* bounded above by a constant. In the second part $\mathrm{Rot}(S^*, T^*)$ is examined and one of two possibilities occurs. The first is that $\mathrm{Rot}(S^*, T^*)$ has size bounded above by a constant; in this case we test each of its members for membership in $\mathrm{Rot}(S, T)$ (by checking whether the image of S is T). The second possibility is that the members of $\mathrm{Rot}(S^*, T^*)$ have a common axis of rotation and, in this case, an extension of Manacher's technique can be used to compute $\mathrm{Rot}(S, T)$. The algorithm can terminate prematurely in either part if it is discovered that S and T are not rotationally congruent.

2.2. *Production of a Small Stable Pair*

In the first of the two parts of the algorithm, we generate stable pairs of decreasing sizes until a suitably small pair is found. The overall structure of this procedure is as follows:

$S^* := S;$ $T^* := T;$
while $|S^*| > K$ **do**
begin
Replace the current stable pair by a new stable pair S^*, T^* of sets of at most half the size
end

In this scheme K is an absolute constant chosen to satisfy certain inequalities which arise in the description and justification of the algorithm.

To find the new stable pair within the body of the **while** loop we employ a number of different tricks but all of them are variants of the following general principle. Suppose we can find a geometric property $\mathscr{P}$, invariant under rotations about the origin, which is possessed by some but not all the points of the current S^*. Thus

$$S' = \{\sigma \in S^* | \sigma \text{ has } \mathscr{P}\}$$

is a proper nonempty subset of S^*. Let

$$T' = \{\sigma \in T^* | \sigma \text{ has } \mathscr{P}\}.$$

If S^* and T^* are congruent then any congruence which maps S^* to T^* must map S' to T' and, in particular, S' and T' must have the same size. Thus, if $|S'| \neq |T'|$, we can immediately conclude that S^* is not congruent to T^* and hence that S is not congruent to T; then the algorithm can terminate. If $|S'| = |T'|$ then both (S', T') and $(S^* - S', T^* - T')$ are stable pairs and, in one of them, the two sets are of size at most half the sizes of S^* and T^*; one of these pairs can then become the new stable pair. We shall refer to this process as *reducing through* $\mathscr{P}$.

The loop body attempts to reduce through $\mathscr{P}$ for a succession of properties $\mathscr{P}$ in the following way. We find a property $\mathscr{P}$ possessed by at least one point of S^*. Then an attempt to reduce through $\mathscr{P}$ leads to one of three outcomes:

(i) The reduction succeeds and we go on to the next iteration of the loop body.

(ii) The algorithm reports that S and T are not congruent (because the sets S' and T' mentioned above have different sizes) and the algorithm terminates.

(iii) The reduction fails because $S' = S^*$ and $T' = T^*$; this means that every point of both S^* and T^* has property $\mathscr{P}$ and, armed with this information, we go on to consider further geometric properties.

To analyse the time spent in the first part of the algorithm we shall show that each attempt to reduce through a property takes a time $O(n^* \log n^*)$, where $n^* = |S^*|$. Since at most four properties are considered it follows that each iteration of the loop body also takes time $O(n^* \log n^*)$; hence the total time spent in the **while** loop is bounded above by a constant multiple of

$$n \log n + n/2 \log n/2 + n/4 \log n/4 + n/8 \log n/8 + \cdots$$

and this expression is of order $n \log n$.

We shall now consider in turn the properties used in attempting to find a new stable pair in some typical iteration of the loop body. Recall that for each property being considered it may be assumed that all the points of both S^* and T^* possess all the previously considered properties.

MODULUS PROPERTY. Choose any point of S^* and let r be its modulus (its distance from the origin). The property of having modulus r is obviously invariant under rotations and we will proceed to consider further properties only if every point of S^* and T^* is of this modulus; that is, in calculating with further properties we may assume that all the points of S^* and T^* lie on the sphere centred at the origin and of radius r. Clearly, the calculations required by this step only require time $O(n^*)$.

The algorithm of [3] can find in time $O(n^* \log n^*)$ the closest pair of points in S^*, but the method of [3] is slightly more general than this. The same divide and conquer technique can be used to find *all* the pairs whose distance apart is minimal even if the minimal distance occurs more than once. By applying this more general form of the algorithm we can define an undirected graph on the set S^*, two points (vertices) being joined by an edge if and only if the distance between them is the minimal distance between pairs of points in S^*. It is easy to see that this graph is planar and that each vertex has degree at most 5. It is most convenient, for our applications, to represent this graph by listing, for each vertex, its adjacent vertices.

VERTEX PROPERTY. For any vertex v in this graph let $V(v)$ be the geometric figure (the *vertex figure*) formed on the sphere by v and its adjacent vertices. Let v_0 be any vertex which is not isolated. Consider the property that a vertex v has vertex figure congruent to $V(v_0)$. Whether a vertex figure has this property can be determined in constant time since a vertex figure has at most 6 points. Therefore we can attempt to reduce through the property of having vertex figure congruent to $V(v_0)$. This calculation requires time $O(n^*)$. It will be necessary to go on to consider further properties only if all the vertex figures in S^* and T^* are congruent. In particular, each vertex has the same degree d, $1 \leq d \leq 5$, and we must now consider the different possibilities for d.

We consider first the case $d = 1$. In this case the n^* points of S^* fall into $n^*/2$ disjoint pairs, each pair being a closest pair. None of the centroids of these pairs can be at the origin, for otherwise the two points would be diametrically opposed and so maximally distant apart; this would imply that $n^* = 2$ which can be ruled out provided that the constant K is taken larger than 2. Therefore we can take the new S^* to be this set of centroids, and define the new T^* similarly.

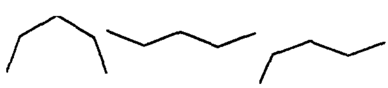
Our discussion of the other values of d requires the following technical proposition whose proof is postponed until the Appendix.

PROPOSITION A. *There exists a monotonic function $f(v)$ with the following property. Let G be any graph whose n^* vertices are on a sphere and for which two vertices are joined if and only if their distance apart is the minimal distance between vertices. Suppose that G has a connected component with v vertices all of the same degree d, with $2 \leq d \leq 5$, whose centroid is at the centre of the sphere. Then $n^* \leq f(v)$.*

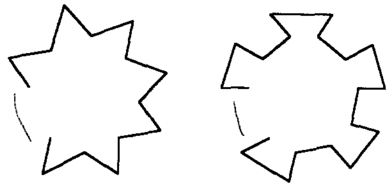
Now we consider the case $d = 2$. In this case the graph on S^* (and on T^*) is a collection of disjoint polygons. Suppose one of these polygons is a triangle. This triangle cannot have its centroid at the origin (for the lemma would give $n^* \leq f(3)$ which cannot hold within the loop body provided that we take $K > f(3)$). Let S', T' be the set of nonzero centroids of the polygons of S^*, T^*. Then S' is nonempty and, if S^* is congruent to T^*, S' is congruent to T'. If $|S'| = |T'|$ we may take S', T' as the next stable pair; or otherwise we conclude that S is not congruent to T. To complete the case $d = 2$ we now have to consider the case that none of the polygons is a triangle.

CHAIN PROPERTY ($d = 2$). For any vertex v we define the chain figure $C(v)$ to be the geometric figure formed by v and the two points before v and the two points after v in the polygon containing v. We fix one vertex v_0 and consider the property that a vertex v has chain figure congruent to $C(v_0)$. We try to reduce through this property. Just as for the vertex property, the calculations require time $O(n^*)$ and further consideration of the case $d = 2$ will only be necessary if all the chain figures of S^* and T^* are congruent.

If the chain figures are all congruent they are congruent to one of



where all the obtuse angles are equal. It follows easily that the polygons are (respectively) regular, or look like



and in all three cases have an axis of symmetry. This axis meets the sphere in two points. The meeting places of these axes can be taken as a new set S^* and we can define T^* similarly. Since every polygon is being replaced by a two-point set the new sets S^*, T^* have size at most half the old sizes.

Finally we have to consider the cases $d = 3, 4, 5$. Recall that a face of a planar graph is a circuit which encloses no other vertex. The *face type* of a vertex v is a sequence $p_1, p_2, \ldots, p_d$ of integers indicating that v is a meeting place of a p_1-gon face, p_2-gon face,..., p_d-gon face (listed in clockwise order). The face types of all the vertices of S^* (and T^*) can be found by an edge traversal in time $O(n^*)$ as follows:

For each vertex, list in clockwise order, the edges out of it, and mark these directed edges as *live*. A *dead* edge (u, v) is one which has been traversed in the direction u to v;
repeat
Find a live edge (u, v);
Starting with this edge follow an edge path in which, on coming to each vertex, we turn to the right to leave it;
On returning to u after, say, p edges, we record a triple $(p$, in-edge, out-edge) for every vertex visited

until all edges are dead.
This traversal visits every directed edge once only and travels clockwise around each face. The resulting triples allow the face type $F(v)$ of every vertex to be determined. For each vertex v we list the triples associated with it in the order (p_1, e_1, e_2), $(p_2, e_2, e_3), \ldots,$ (p_{d-1}, e_{d-1}, e_d), (p_d, e_d, e_1).
Then the face type of v is $(p_1, p_2, \ldots, p_d)$.

FACE PROPERTY ($d = 3, 4, 5$). Let v_0 be any vertex. The property that a vertex has face type equal to $F(v_0)$ is invariant under congruence and so we can try to reduce through this property. As usual, we only need to continue the analysis further if all vertices of S^* and T^* have the same face type. If all the vertices do have the same face type we have enough information that the graphs on S^* and T^* are severely restricted.

Consider an arbitrary connected component of S^* or T^* with q vertices. Since the degree of each vertex is d this component has $qd/2$ edges. Suppose that the face type of each vertex is $(p_1, \ldots, p_d)$. Then the total number of faces in the component is

$$q(1/p_1 + 1/p_2 + \cdots + 1/p_d).$$

Therefore, by Euler's formula,

$$q + q(1/p_1 + 1/p_2 + \cdots + 1/p_d) = qd/2 + 2$$

or

$$1/p_1 + 1/p_2 + \cdots + 1/p_d = d/2 - 1 + 2/q. \qquad (\dagger)$$

PROPOSITION B. *The diophantine equation* ($\dagger$) *has only three infinite parametric families of solutions*:

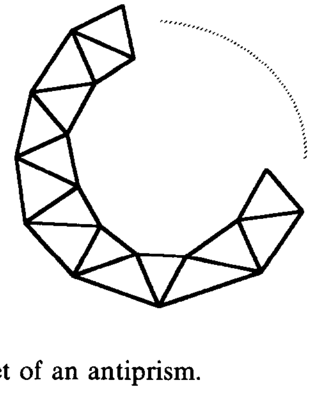
(1) $d = 4,\ p_1 = p_2 = p_3 = 3,\ p_4 = q/2,$

(2) $d = 3,\ p_1 = p_2 = 4,\ p_3 = q/2,$

(3) $d = 3, \{p_1, p_2\} = \{3, 6\}, p_3 = q/2,$

and a finite number of sporadic solutions.

The calculations which yield these solutions are very similar to those which arise in the classification of the Archimedean and Platonic solids, and the proof is delayed until the Appendix. In fact, slightly more argument proves that each sporadic solution either does not correspond to an actual graph or corresponds uniquely to an Archimedean or Platonic solid. We do not need this fact but we do require the fact that the parametric solution (1) corresponds to an antiprism. Indeed it is easily seen by direct construction that solution (1) drawn in the plane is



which is the planar net of an antiprism.

When drawn on the sphere all the triangles must be equilateral and hence equiangular. This forces the antiprism to have its two $q/2$-gon faces plane and parallel. The antiprism therefore has an easily computable axis and we may calculate the two meeting points of this axis with the sphere. Thus when S^* (and T^*) is a collection of antiprisms we can take these pairs of meeting points as the new sets S^*, T^*.

In the case of solution (2) each component has $q/2$ quadrilaterals (in fact these components are prisms). Lemma 2 (in the Appendix) ensures that the quadrilaterals do not have centroid at the origin (so long as $K > 9$) and for the new S^*, T^* we take the centroids of the quadrilaterals of S^*, T^*.

Solution 3 is impossible if $q > 12$, and so may be included among the sporadic solutions. For if the face type of a vertex is $(3, 6, q/2)$ (to within a cyclic shift) then a neighbouring vertex must have face type $(6, 3, q/2)$ (again to within a cyclic shift) and so not all the vertices have the same face type.

For the sporadic solutions let v be the maximum component size. Then, by Proposition A, $n^* \leq f(v)$ and so, provided we take $K > f(v)$, the sporadic solutions do not occur in this part of the algorithm.

2.3. *Examination of* Rot(S^*, T^*)

This part of the algorithm divides into two cases

(i) $|S^*| = 1$, or $|S^*| = 2$ and S^* consists of two diametrically opposed points on the sphere,

(ii) $|S^*| > 2$, or $|S^*| = 2$ and the two points of S^* are not diametrically opposed.

In the latter case we can find two points α, β of S^* which are not diametrically opposed. Any rotation about the origin is completely determined by its effect on the points α, β. In particular each member of Rot(S^*, T^*) is determined uniquely by the images of α, β and, since $|T^*|$ is bounded, Rot(S^*, T^*) is a subset of some known bounded set Δ. However, Rot(S, T) is contained in Rot(S^*, T^*) and so may be determined by checking each element of Δ for whether it maps S onto T; these checks require time $O(n \log n)$.

Case (i) requires slightly more discussion, and here we use a generalisation of the technique used by Manacher [2] in his treatment of the 2-dimensional case. When the conditions of (i) hold, the single point (or two diametrically opposed points) of S^* determines an axis through the origin. Any congruence which maps S^* to T^* must transform the axis of S^* to the corresponding axis of T^*; and any two such congruences differ only by a rotation about one of these axes. Therefore we may apply a transformation to map the axis of S^* to the axis of T^* and thereafter assume that these two axes are the same. Since $\mathrm{Rot}(S, T)$ is a subset of $\mathrm{Rot}(S^*, T^*)$ we may obtain all the congruences of $\mathrm{Rot}(S, T)$ by determining the rotations about the fixed common axis of S^* and T^* which map S to T.

By a change of coordinate system we may take the common axis of S^* and T^* to be the z axis. We express every point of S, T in cylindrical coordinates (r, θ, z) and list the points of S and T in increasing order of argument θ (and for points with the same argument, in increasing order of r coordinate). Let

$$S = \{(\sigma_0, \theta_0, z_0), \ldots, (\sigma_{n-1}, \theta_{n-1}, z_{n-1})\}, \qquad \theta_0 \le \theta_1 \le \cdots \le \theta_{n-1},$$
$$T = \{(\tau_0, \phi_0, w_0), \ldots, (\tau_{n-1}, \phi_{n-1}, w_{n-1})\}, \qquad \phi_0 \le \phi_1 \le \cdots \le \phi_{n-1}.$$

A necessary and sufficient condition for there to be a rotation which maps S onto T is that, for some j,

$$\sigma_k = \tau_{k+j}, \qquad z_k = w_{k+j}, \qquad \theta_{k+1} - \theta_k = \phi_{k+1} - \phi_k,$$
$$k = 0, 1, \ldots, n-1$$

(where here, and subsequently, all subscripts are reduced modulo n and the last equation is interpreted modulo 2π). If we let

$$\Sigma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \qquad \text{and} \qquad T = (\beta_0, \beta_1, \ldots, \beta_{n-1}),$$

where

$$\alpha_k = (\sigma_k, z_k, \theta_{k+1} - \theta_k) \qquad \text{and} \qquad \beta_k = (\tau_k, w_k, \phi_{k+1} - \phi_k)$$

we can restate this condition as follows: there is a one-to-one correspondence between rotations which map S onto T and integers j, $0 \le j < n$, such that

$$\alpha_k = \beta_{k+j}, \qquad k = 0, 1, \ldots, n-1.$$

Integers j satisfying this criterion correspond precisely to occurrences of the "pattern" Σ within the "text" $\beta_0\beta_1 \cdots \beta_{n-1}\beta_0 \cdots \beta_{n-2}$ and they may be found in time $O(n)$ by the pattern matching algorithm of [1].

Appendix

We derive the proof of Propositions A and B used in Section 2.2 in a series of lemmas. I thank Dr. Ralph Martin for the proof of the first lemma and thereby substantially shortening my original argument.

LEMMA 1. *Every closed curve of length less than $2\pi r$ lying on a sphere of radius r lies entirely within a hemisphere.*

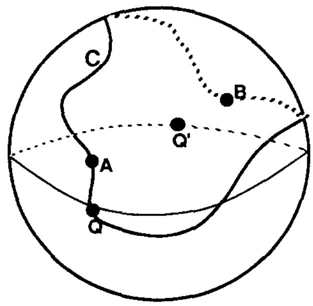


FIGURE 1

Proof. Suppose that a closed curve C has length $d < 2\pi r$. Choose two points A, B which are distance $d/2$ apart measured along the curve. Let P be the midpoint of a great circle segment joining A to B. We shall show that the hemisphere which has P as its pole contains the curve.

Suppose that it does not, in which case the curve intersects the boundary of the hemisphere at a point Q. Now consider a different curve QBQ' (see Fig. 1); the part from Q to B is the same as on the curve C and the part from B to Q' is obtained by rotating the segment AQ on the curve C through an angle π about the axis of the hemisphere (note that B is the image of A under this rotation).Then

$$\begin{aligned}\text{length}(QBQ') &= \text{length}(QB) + \text{length}(BQ')\\ &= \text{length}(QB) + \text{length}(AQ)\\ &= \text{length}(AB)\\ &= d/2 < \pi r.\end{aligned}$$

However, Q and Q' are at opposite ends of a diameter and so any curve which joins them lying on the sphere must have length at least πr. This contradiction shows that the hemisphere contains the curve C.

LEMMA 2. *Let S^* be a graph with vertices on a sphere with an edge between two vertices if the distance between them is the smallest distance m that occurs between vertices. Suppose also that there is a polygon with t vertices also on the sphere whose centroid is at the centre of the sphere, and whose successive vertices are distance m apart. Then*

$$|S^*| \leq 16/\left(\sin^2\pi/t\ \left(4 - \sin^2\pi/t\right)\right).$$

Proof. Let r be the radius of the sphere. By joining successive vertices of the polygon by segments of great circles we obtain a curve on the sphere whose centroid is at the origin. In particular the curve (and also the polygon) does not lie within a hemisphere. Hence the length of the curve is at least $2\pi r$ and each great circle segment has length at least $2\pi r/t$. Therefore consecutive points on the polygon subtend a central angle of at least $2\pi/t$ and so $m \geq 2r \sin \pi/t$.

If we surround every point of S^* by a ball of radius $r \sin \pi/t$ none of these balls can overlap. Each of them intersects the sphere in a circle of radius

$$\tfrac{1}{2} r \sin \pi/t \sqrt{(4 - \sin^2 \pi/t)}$$

and so these circles enclose areas on the sphere of at least

$$\tfrac{1}{4} r^2 \sin^2 \pi/t (4 - \sin^2 \pi/t).$$

Thus

$$|S^*| \cdot \tfrac{1}{4} r^2 \sin^2 \pi/t (4 - \sin^2 \pi/t) \leq 4\pi r^2$$

from which the result follows.

LEMMA 3. *Let S^* be a graph with vertices on a sphere with an edge between two vertices if the distance between them is the smallest distance that occurs between vertices. Suppose the graph has a connected component with v vertices all of degree d whose centroid is at the centre of the sphere. Then*

$$|S^*| \leq 16/(\sin^2 \pi/vd(4 - \sin^2 \pi/vd)).$$

In particular, since $d \leq 5$, $|S^*|$ is bounded in terms of the size of the component.

Proof. The connected component has a closed (self-intersecting) path visiting each vertex d times and each edge twice, once in each direction. This path defines a (degenerate) polygon with vd vertices whose centroid is at the centre of the sphere, and successive vertices on it are distance m apart. Now apply Lemma 2.

The last lemma suffices to prove Proposition A, and we now turn to the proof of Proposition B.

LEMMA 4. *Let $1/x + 1/y = k + 2/q$ where k is some explicit positive constant. If k is not of the form $1/t$ for some integer t, there are only a finite number of solution triples (x, y, q) in positive integers. If k has the form $1/t$ there is, in addition, only the infinite families of solutions $(x, y, q) = (t, q/2, q), (q/2, t, q)$ for all even q.*

Proof. Assume $x \leq y$. Then $2/x \geq 1/x + 1/y = k + 2/q > k$. Therefore $x < 2/k$ and there are only a finite number of possibilities for x.

Consider one of these possibilities $x = x_0$, where $x_0 \neq t$, in the case $k = 1/t$. Then $1/y = k - 1/x_0 + 2/q = j + 2/q$, where j is some positive or negative (but nonzero) constant. If j is positive we have $1/y > j$ or $y < 1/j$, so that y is also bounded, and there are only a finite number of possibilities for y (and so a finite number of possibilities for q). If j is negative then $2/q = 1/y - j > -j$; thus $q < -2/j$, and again there are only a finite number of possibilities for q and y. The only case not covered by this argument is where k has the form $1/t$ and $x_0 = t$; clearly $y = q/2$ giving the infinite family in the statement of the lemma.

We now consider, in turn, the values of d in Eq. (†).

(1) ($d = 3$) $1/a + 1/b + 1/c = 1/2 + 2/q$. Assume $a \leq b \leq c$. Not all of a, b, c exceed 5, for otherwise

$$\tfrac{1}{2} = \tfrac{1}{6} + \tfrac{1}{6} + \tfrac{1}{6} \geq 1/a + 1/b + 1/c = \tfrac{1}{2} + 2/q > \tfrac{1}{2}.$$

Hence we can take $a = 3$, 4 or 5.

1.1 ($a = 5$) Here $1/b + 1/c = \frac{3}{10} + 2/q$, and Lemma 4 shows this has but finitely many solutions.

1.2 ($a = 4$) Here $1/b + 1/c = \frac{1}{4} + 2/q$. Lemma 4 shows that the only solutions are $(b, c, q) = (4, q/2, q), (q/2, 4, q)$ and finitely many others.

1.3 ($a = 3$) Here $1/b + 1/c = \frac{1}{6} + 2/q$. Lemma 4 shows that the only solutions are $(b, c) = (6, q/2, q), (q/2, 6, q)$ and finitely many others.

(2) ($d = 4$) $1/a + 1/b + 1/c + 1/d = 1 + 2q$. Assume $a \leq b \leq c \leq d$. Not all of a, b, c, d exceed 3, for otherwise

$$1 = \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} \geq 1/a + 1/b + 1/c + 1/d = 1 + 2/q > 1.$$

Hence we may take $a = 3$. Then

$$1/b + 1/c + 1/d = \tfrac{2}{3} + 2/q.$$

Not all of b, c, d can exceed 4, for otherwise

$$\tfrac{3}{5} = \tfrac{1}{5} + \tfrac{1}{5} + \tfrac{1}{5} \geq 1/b + 1/c + 1/d = \tfrac{2}{3} + 2/q > \tfrac{2}{3}.$$

Hence $b = 3$ or 4.

2.1. ($b = 4$) Here $1/c + 1/d = \frac{5}{12} + 2/q$ and Lemma 4 shows this has only finitely many solutions.

2.2 ($b = 3$) Here $1/c + 1/d = \frac{1}{3} + 2/q$ and apart from finitely many solutions we have, again by Lemma 4, only $(c, d) = (3, q/2, q), (q/2, 3, q)$.

(3) $(d = 5)$ $1/a + 1/b + 1/c + 1/d + 1/e = \frac{3}{2} + 2q$. Assume $a \leq b \leq c \leq d \leq e$. Not all of a, b, c, d, e exceed 3 for otherwise

$$\tfrac{5}{4} = \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} \geq 1/a + 1/b + 1/c + 1/d + 1/e = \tfrac{3}{2} + 2/q > \tfrac{3}{2}$$

Hence we may take $a = 3$ and obtain

$$1/b + 1/c + 1/d + 1/e = \tfrac{7}{6} + 2/q.$$

Again, not all of b, c, d, e can exceed 3 for otherwise

$$1 = \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} \geq 1/b + 1/c + 1/d + 1/e = \tfrac{7}{6} + 2/q > \tfrac{7}{6}.$$

Hence we may take $b = 3$ and obtain

$$1/c + 1/d + 1/e = \tfrac{5}{6} + 2/q.$$

Again, not all of c, d, e can exceed 3 for otherwise

$$\tfrac{3}{4} = \tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{4} \geq 1/c + 1/d + 1/e = \tfrac{5}{6} + 2/q > \tfrac{5}{6}.$$

Thus $c = 3$. We now have

$$1/d + 1/e = \tfrac{1}{2} + 2/q.$$

Once more it is clear that not both d, e exceed 3 and so $d = 3$. Then $1/e = 1/6 + 2/q > \frac{1}{6}$ and, since $e < 6$, there are only finitely many solutions.

PROPOSITION B. *The diophantine equation* (†) *only has solutions*

(1) $d = 3$, $\{p_1, p_2, p_3\} = \{4, 4, q/2\}$,

(2) $d = 3$, $\{p_1, p_2, p_3\} = \{3, 6, q/2\}$,

(3) $d = 4$, $\{p_1, p_2, p_3, p_4\} = \{3, 3, 3, q/2\}$,

together with finitely many others.

REFERENCES

1. D. E. KNUTH, J. H. PRATT, AND V. R. MORRIS, Fast pattern matching in strings, *SIAM J. Comput.* **6** (1977), 323–350.
2. G. K. MANACHER, An application of pattern matching to a problem in geometric complexity, *Inform. Process. Lett.* **5** (1976), 6–7.
3. E. M. REINGOLD, On the optimality of some set algorithms, *J. Assoc. Comput. Mach.* **19** (1972), 649–659.
4. M. I. SHAMOS AND J. L. BENTLEY, Divide and conquer in multi-dimensional space, *in* "Proceedings 8th Annual ACM Symposium on Theory of Computing," 1976, pp. 220–230.
5. K. SUGIHARA, An $n \log n$ algorithm for determining the congruity of polyhedra, *J Comput. System Sci.* **29** (1984), 36–47.