Discrete Comput Geom 13:123–140 (1995)

Discrete & Computational
Geometry

Reconfiguring Closed Polygonal Chains in Euclidean d-Space*

W. J. Lenhart[1] and S. H. Whitesides[2]

[1] Department of Computer Science, Williams College,
Williamstown, MA 01267, USA
lenhart@cs.williams.edu

[2] School of Computer Science, McGill University,
3480 University Street, Montréal H3A 2A7, Canada
sue@cs.mcgill.ca

Abstract. Consider the problem of moving a closed chain of n links in two or more dimensions from one given configuration to another. The links have fixed lengths and may rotate about their endpoints, possibly passing through one another. The notion of a "line-tracking motion" is defined, and it is shown that when reconfiguration is possible by any means, it can be achieved by $O(n)$ line-tracking motions. These motions can be computed in $O(n)$ time on real RAM. It is shown that in three or more dimensions, reconfiguration is always possible, but that in dimension two this is not the case. Reconfiguration is shown to be always possible in two dimensions if and only if the sum of the lengths of the second and third longest links add to at most the sum of the lengths of the remaining links. An $O(n)$ algorithm is given for determining whether it is possible to move between two given configurations of a closed chain in the plane and, if it is possible, for computing a sequence of line-tracking motions to carry out the reconfiguration.

1. Introduction

The problem of reconfiguring chains of n links under various conditions has been considered from an algorithmic point of view in [5], [7]–[11], and [16]. In particular, these papers present polynomial-time algorithms for planar motion-planning problems that have an unbounded number of degrees of freedom. While there are general techniques [13], [1] for solving motion-planning problems having a bounded number of degrees of freedom in polynomial time, problems

* The research of the second author was partially supported by an NSERC operating grant.

having an unbounded number of degrees of freedom are often at least NP-complete. Even problems involving chains of n links or trees of n links can be NP-complete, NP-hard [5], [16], [17], or P-space hard [7]. See [12] for a P-space hardness result for tree-like linkages and [4] for a P-space hard problem for graph-like linkages. Hence it is of interest to find examples of motion-planning problems that can be solved quickly despite having an unbounded number of degrees of freedom. This paper contributes such an example. (For collected works on algorithmic motion planning, see [14] and [15].)

The study of linkages has also been pursued in other disciplines. For example, topologists study the properties of the configuration space of linkages. Each configuration in physical space is regarded as a point in some configuration space, and one asks for topological properties of this set of points, such as the number of its connected components. (See, for example, [6] and [3].) While the focus of our research is on planning motions that can be carried out physically, we obtain some topological results as a by-product of our algorithm design. Some of these results also have direct topological proofs. The study of linkages is also of interest to those investigating the design of large molecules. See, for example, [2].

In this paper we give an $O(n)$ algorithm for moving a closed chain of links embedded in $d \geq 3$ dimensions from any given initial configuration to any given final configuration, where the links are allowed to pass through one another during the motion. For dimension $d = 2$, we show that a closed chain can always be moved between any given pair of configurations if and only if the lengths of its second and third longest links sum to at most half the sum of the lengths of *all* of the links. (Note: there may be several links of the same length; by "second and third longest links" we mean the links that would appear in positions two and three of a list of the links sorted by decreasing length.) We also give an algorithm that uses "line-tracking" motions, defined in Section 2, to reconfigure a closed chain of links provided that reconfiguration is possible by some arbitrary motion. Our algorithms in $d = 2$ and $d \geq 3$ dimensions require $O(n)$ line-tracking motions for an n-link chain, and the descriptions of these motions can be calculated in $O(n)$ time. The model of computation used for all algorithms is real RAM.

In order to state our results, we need to introduce some terminology. An *open chain* is a weighted abstract graph Q having vertices (joints) $\{v_0, v_1, \ldots, v_n\}$ and edges (links) $\{L_1, \ldots, L_n\}$, where $L_i = (v_{i-1}, v_i)$ and each edge L_i has positive weight (length) l_i. A *configuration* of a chain $Q = L_1, \ldots, L_n$ is a polygonal curve (possibly self-intersecting) that consists of n consecutive segments of lengths $l_1, \ldots, l_n$, respectively. A *closed chain* is defined similarly to an open chain except that $v_0 = v_n$. Hence a configuration of a closed chain is just a closed polygonal curve. As Lemma 3.1 shows, a closed chain admits configurations if and only if there is no link L_i having length greater than the sum of the lengths of the remaining links. An *arm* is an open chain in which a fixed location has been associated with v_0.

The distance between two points x and y is denoted $d(x, y)$, and the line determined by two points x and y is denoted $l(x, y)$. Also, given an open or closed chain, the *subchain from joint* $u = v_i$ *to joint* $w = v_j$ is the open chain consisting of joints $v_i, v_{i+1}, \ldots, v_j$ with subscripts assumed to be increasing and taken mod $n + 1$. The same convention holds for summations.

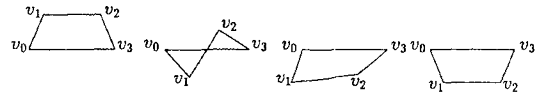

Fig. 1. Inverting a closed chain in two dimensions. (v_0, v_1) rotates clockwise about v_0 as (v_2, v_3) rotates first clockwise, then counterclockwise about v_3; then (v_0, v_1) rotates counterclockwise about v_0 as (v_2, v_3) continues to rotate counterclockwise about v_3.

Definition 1.1. Two configurations of a chain Q are *equivalent* if one configuration can be continuously moved to the other.

In mathematical terms this means that there is a homotopy between the two polygonal curves having the property that the links remain straight and their lengths are preserved *throughout* the homotopy. Note that links are allowed to intersect during the motion. Clearly, the definition of equivalence gives an equivalence relation on the set of configurations of a chain.

Definition 1.2. A configuration of a closed chain Q in $d \geq 2$ dimensions is *invertible* if it is equivalent to its mirror image (with respect to some arbitrary hyperplane).

Figures 1 and 2 give examples of invertible and noninvertible closed chains in the plane.

Our results are organized into sections as follows. Section 2 discusses the line-tracking motions used in the algorithms and in the constructive proofs. Section 3 introduces "standard triangular form" for a closed chain and proves that any configuration of a closed chain can be moved to a standard triangular form. From this it can be immediately concluded that a closed chain has only one equivalence class of configurations in $d \geq 3$ dimensions. Thus invertibility for closed chains is only an issue in the plane. Section 4 handles the case of the plane, where reconfiguration is not always possible, and shows that a closed chain of links in the plane has at most two equivalence classes of configurations. If the chain satisfies the property that the lengths of the second and third longest links sum to at most half the sum of the lengths of all of the links (or, equivalently, sum to no more than the sum of the lengths of the remaining links), then the chain has just one

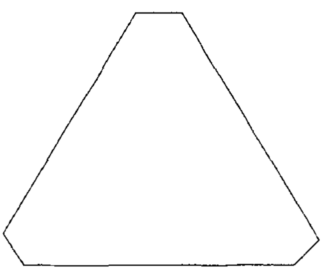

Fig. 2. A six-link chain that is not invertible.

equivalence class of configurations: it is possible to move between any given pair of configurations. If this property does not hold (see Fig. 2 for an example), then the chain has exactly two equivalence classes of configurations, and the configurations in the one class are the mirror images of the configurations in the other class. In this case it is possible to move between two given configurations if and only if they lie in the same equivalence class. Section 5 summarizes the results and lists some open problems.

2. Simple Motions and Line-Tracking Motions

Motion-planning algorithms must compute unambiguous descriptions of motions. To achieve this, it is useful to define one or more kinds of simple motion steps, so that complicated motions can be described as a sequence of the simple ones. Of course, the simple motions chosen should not be limiting: it should be possible to carry out any reconfiguration in terms of the simple motions available to the algorithm. Here is a list of criteria, based on [5], for "good" simple motions:

Criteria:

1. The description of the motion should uniquely determine the geometric movement of all parts of the linkage.
2. The motion should be one whose description can be computed.
3. If the angle at a joint changes, it should change monotonically. In other words, a motion in which a given angle increases and then decreases should be regarded as a combination of simpler motions.

The criteria given above allow for many kinds of motions. Before defining the types of motion which are used in this paper, we make a few remarks concerning the region of points reachable by the end of an arm.

Definition 2.1. Let Q be an arm with joints $v_0, v_1, \ldots, v_n$, where the position of v_0 is fixed. The *reachable region* R_j, $1 \leq j \leq n$, of v_j is the set of all points that v_j can reach.

Remark. As mentioned in [5], it is easy to show that R_j is either a ball of radius $l = \sum_{i=1}^{j} l_i$ centered at v_0 or an "annulus" centered at v_0, that is, a ball of outer radius l centered at v_0 minus the interior of a ball of radius l', $0 < l' \leq l$, centered at v_0. Moreover, R_j is an annulus if and only if there is a link L_k, $k \leq j$, such that

$$l_k > \sum_{i \leq j, i \neq k} l_i,$$

in which case the annulus has inner radius

$$l' = l_k - \sum_{i \leq j, i \neq k} l_i.$$

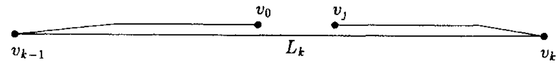

Fig. 3. v_0 is as close as possible to v_j.

To see this, observe that any configuration of the first j links that places v_j as close as possible to v_0 consists, for some $k \leq j$, of straight, possibly empty, chains of links $L_1, L_2, \ldots, L_{k-1}$ and $L_{k+1}, L_{k+2}, \ldots, L_j$ lying along link L_k. See Fig. 3.

We now define a motion (called *simple elbow bending*) that satisfies Criteria 1–3 above and that is an essential ingredient of the definition of line-tracking motion. The elbow-bending motion applies to an open chain of links.

Definition 2.2. An *elbow* $E(x, y, z)$ is a two-link arm consisting of joints x, y, and z, where the location of x is fixed. The angle at the *elbow joint* y may change, and the entire elbow may rotate about the fixed joint x. An *elbow motion* consists of moving z in a straight line from its initial location to a specified final location.

For concreteness, if $d \geq 3$, the motion occurs in the plane determined by x and the line along which z moves. If necessary, y is first moved into that plane, by rotating its two links about x and z, respectively.

There are two special types of configuration for an elbow $E(x, y, z)$. The elbow is said to be *folded* if x is as close to z as possible and *straightened* if x is as far from z as possible. Similarly, the action of bringing z closer to x is called *folding* the joint y, and the action of moving z away from x is called *straightening* y.

Observation 2.1. It is easy to show that if $\overline{pq}$ is a line segment contained within the reachable region of the free end z of an elbow $E(x, y, z)$, and if z is at location p, then the elbow can be moved so that z tracks along the entire segment to q in such a way that the angles at joints x and y will change either monotonically or unimodally.

Observation 2.2. Given a closed segment $\overline{pq}$ contained in the interior of the reachable region of the free end z of an elbow $E(x, y, z)$, the elbow motion from p to q is determined by the initial configuration of the arm. If some point on $\overline{pq}$ other than q lies on the boundary of the reachable region of z, then additional information can be given to specify in which direction the elbow joint y should move. Hence elbow motions satisfy Criteria 1 and 2 above.

Observation 2.3. During an elbow motion that moves z along a line segment within its reachable region, the joint angle at y might decrease (though not necessarily fold completely), then increase (though not necessarily straighten completely). Also, in the plane of the elbow motion, the first link L_1 might rotate first in one sense, then in the opposite sense (so the angle formed at x between L_1

and a reference line both increases and decreases). Thus an elbow motion does not always satisfy Criterion 3 above. This prompts the following definition:

Definition 2.3. A *simple elbow motion* is an elbow motion in which the joint angles at x and y each change monotonically.

Clearly, every elbow motion can be decomposed into at most a constant number (in fact, three) of simple elbow motions that meet Criteria 1–3.

Next, we define line-tracking motions. These are combinations of simple elbow motions that cause at most five joint angles to change simultaneously. The idea is that parts of a chain Q should play the role of two elbows with a common free end; the two elbows cooperate to move their common free end along a line. The links connecting the fixed joints of the two elbows do not move.

Definition 2.4 (see Fig. 4). Let $Q = v_0, \ldots, v_n$ be a closed chain, and let v_a, v_b, v_c, v_d, and v_e be five consecutive joints of Q as they would appear in order around the chain (note that v_a and v_e are the same joint if $n = 4$). Let M be a line containing v_c. A *line-tracking* motion is one such that:

1. v_c moves in one direction along M.
2. The locations of joints v_a and v_e remain fixed throughout the motion, as does the subchain from v_e to v_a.
3. The two links between v_a and v_e act as an elbow $E(v_a, v_b, v_c)$ undergoing an elbow motion; similarly, the two links between v_e and v_c act as an elbow $E(v_e, v_d, v_c)$ undergoing an elbow motion. The planes of these elbow motions need not be the same in dimension $d > 2$.

Definition 2.5. A *simple line-tracking* motion is a line-tracking motion in which both $E(v_a, v_b, v_c)$ and $E(v_e, v_d, v_c)$ move in simple elbow motions.

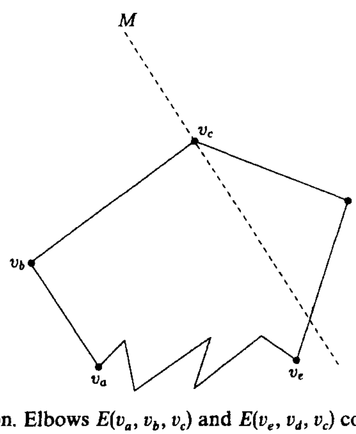

Fig. 4. Line-tracking motion. Elbows $E(v_a, v_b, v_c)$ and $E(v_e, v_d, v_c)$ cooperate to move v_c along line M; the two elbows need not determine the same plane in dimension $d > 2$.

Clearly, any line-tracking motion can be decomposed into at most a constant number (independent of n) *simple* line-tracking motions. Specifying the initial configuration, the line M, and the stopping position for the joint v_c can be regarded as specifying the motion completely. (From this, specifications for the constituent simple motions can be computed.)

3. Standard Triangular Form

This section defines a "standard triangular form" for a closed chain and proves that any configuration of a chain in $d \geq 2$ dimensions can be moved to a standard triangular form.

Definition 3.1. Consider a configuration of a closed chain of n links in $d \geq 2$ dimensions. Let m be the total length of the chain, let i be the smallest index such that L_i has maximum length, and let j be the index such that

$$\sum_{k=i}^{j-1} l_k \leq m/2 \quad \text{but} \quad \sum_{k=i}^{j} l_k > m/2.$$

Now let $x = v_{i-1}$, $y = v_{j-1}$, and $z = v_j$. Then the chain is said to be in *standard triangular form* if all joints other than x, y, and z have joint angle equal to π.

Note that standard triangular form is unique up to isometry. Note also that the configuration has the shape of a (possibly degenerate) triangle with vertices x, y, and z. See Fig. 5 for an example of a configuration in standard triangular form.

Lemma 3.1. *The following are equivalent for a closed n-link chain L:*

1. *L admits a standard triangular form.*
2. *L admits some configuration.*
3. *L contains no link the length of which is greater than the sum of the lengths of the remaining links.*

Proof. Clearly 1 implies 2 and 2 implies 3. To prove that 3 implies 1 we describe how to assign locations to the joints. Compute m, i, j, x, y, and z as in Definition 3.1; this can easily be done in $O(n)$ time. Now, place x at the origin and z on the

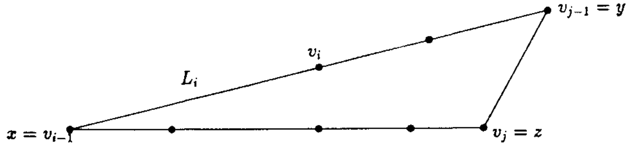

Fig. 5. A configuration in standard triangular form.

positive x-axis at a distance from the origin equal to the sum of the lengths of the links in the chain section from v_j to v_{i-1}. Then place y at an intersection point of two circles in the (x, y)-plane: the circle, centered at z, of radius equal to the length of link (v_{j-1}, v_j) and the circle, centered at x, of radius equal to the sum of the lengths of the links between v_{i-1} and v_{j-1}. Note that these circles must have nonempty intersection: The circle centered at x cannot contain the circle centered at z by the choice of v_{j-1}. The circle centered at z cannot contain the circle centered at x: if it did, then link (v_{j-1}, v_j) would be longer than the sum of the lengths of all the other links, contradicting that the closed chain admits a configuration. Finally, the two circles cannot lie in each other's exteriors, as the sum of their radii is greater than $m/2$ and hence greater than the distance between x and z. □

We now show that any configuration of a closed chain can be moved to a standard triangular form using a linear number of a line-tracking motions. The proof consists of an algorithm which first moves the configuration into a (possibly degenerate) triangular shape and then adjusts this triangular configuration until it is in standard triangular form. Before presenting the algorithm, which appears in Fig. 6, we describe some of the operations the algorithm performs.

If C is a configuration of a closed chain Q and if v is a straightened joint of C, to *remove v from C* means to replace the two links adjacent to v with a single link whose length equals the sum of the lengths of the two links adjacent to v. Note that the joint is not removed from Q, only from C. Suppose C is a configuration of Q and that some remove operations are performed on C and suppose further that v is a joint of Q that was removed from C. *Reinserting v into C* is defined as follows: Let w be the first joint in Q after v such that $w \in C$, and let u be the last joint in Q before v such that $u \in C$; thus there is a link in C from u to w. Replace this link in C with a link from u to v of length equal to the sum of the lengths of the links in Q from u to v along with a link from v to w of length equal to the sum of the lengths of the links in Q from v to w.

Note that the shape of C (and its length) remain unchanged by remove and reinsert operations; only the number of links changes. Note also that a remove operation can be performed in constant time and that a reinsert operation can be performed in time proportional to the size of Q, since all that is needed is to compute the lengths of the two links added. Moreover, reinserting into C all joints of Q not in C can also be accomplished in time proportional to the size of Q by traversing C and Q simultaneously.

Theorem 3.1. *Any configuration of a closed n-link chain in $d \geq 2$ dimensions can be moved to a standard triangular form with $O(n)$ simple line-tracking motions whose descriptions can be computed in $O(n)$ time.*

Proof. The proof consists of verifying the correctness and analyzing the time complexity of procedure *standard_triangular_form* presented in Fig. 6. Let Q be a closed n-link chain. The first phase of the algorithm repeatedly straightens a joint of configuration C of the closed chain Q and then removes the straightened

```
procedure standard_triangular_form(Q, C)
{ Q is a closed chain and C is a configuration of Q; }
{ x, y and z are the three joints of Q given in Definition 3.1. }

    { phase 1: Move C to a triangular shape. }
    while C has at least four joints
        choose v_a, v_b, v_c, v_d, v_e to be any five consecutive joints of C
        { If C has only four joints then v_a = v_e. }
        if v_a and v_e have different locations then
            choose M to be the line through v_c perpendicular to l(v_a, v_e)
        else
            choose M to be any line through v_c
        endif
        move v_c away from both v_a and v_e along M with a line tracking
        motion until v_b or v_d straightens
        if v_b straightens then remove v_b from C else remove v_d from C
    endwhile
    { C is now a possibly degenerate triangle T. }

    { phase 2: Reshape into standard triangular form. }
    if x is not a joint of T then adjust_triangle(T, x)
    if y is not a joint of T then adjust_triangle(T, y)
    if z is not a joint of T then adjust_triangle(T, z)
    reinsert into T all joints of Q not in T
end standard_triangular_form

procedure adjust_triangle(T, v)
{ T is a possibly degenerate triangle; v is a joint of Q not in T. }
    reinsert v into T
    find the joint w in T not adjacent to v
    if w and v have different locations then
        choose M to be the line l(w, v)
    else
        choose M to be any line through v
    endif
    move v away from w along M with a line tracking
    motion until some joint u straightens
    remove u from T
end adjust_triangle
```

Fig. 6. The algorithm for moving to standard triangular form.

joint from C. To straighten a joint, any five consecutive joints, say v_a through v_e, are selected ($v_a = v_e$ if C has only four joints). Now v_c is moved by a line-tracking motion along a line M chosen so that the distance from v_c to each of v_a and v_e increases monotonically. Thus the angles at both v_b and v_d are monotonically increasing throughout the motion so that either v_b or v_d will straighten. This may require more than one (but at most a constant number of) simple line-tracking motions because the angles at joints v_a and v_e may not change monotonically. The straightened joint and its two links are then replaced by a single link of length equal to the sum of the lengths of the two removed links. Repeating this process transforms the original configuration into a configuration T having a (possibly degenerate) triangular shape. Only $O(n)$ simple line-tracking motions are used.

Let x, y, and z denote the joints of Q specified in the definition of standard triangular form. If these happen to be the three joints of T, then the algorithm reinserts into C the $n - 3$ joints of Q that were removed, giving a configuration of Q in standard triangular form. This can be done in $O(n)$ time. If the three joints of T are not x, y, and z then the algorithm adjusts T (in $O(n)$ time) so that T has joints x, y, and z. This adjustment is described below.

If x is not in T, then x is reinserted into T using procedure *adjust_triangle*. This procedure first adds x back to T and then moves x away from that joint w of T not adjacent to x until some joint of T straightens. The straightened joint of T is now removed. The same procedure is invoked, if necessary, to reinsert y and z, in that order, so that x, y, and z become the joints of T.

The only way that reinserting y could cause x to be removed from T is if x straightens as y is moved. For this to occur, however, y must have been reinserted into the interior of one of the links adjacent to x. This means that this link of T had length greater than $m/2$, which contradicts the definitions of x, y, and z. Thus reinserting y into T using procedure *adjust–triangle* does not cause x to be removed.

Now consider applying *adjust–triangle* to T in order to reinsert z. Let t be the third joint of T before z is reinserted. By definition of x and y, the chain in Q from x to y does not contain t. Clearly, z must be inserted into the link from y to t. Thus z will be moved away from x in procedure *adjust–triangle* and so x will not straighten. Also, by definition of x, y, and z, y cannot straighten. □

The proof of the preceding theorem shows how to move any configuration in dimension $d \geq 2$ to a triangle and then to standard triangular form. Another approach would be to move the initial configuration to a planar one, and then to move this to a standard triangular form within the plane. This could be done by $O(n)$ motion steps involving rotation of a rigid subsequence of links about a line through the endpoints of the subsequence. However, as the proof of Theorem 3.1 shows, such a planarization step is not necessary and would only increase the number of moves.

Theorem 3.2. *Any two configurations of a closed n-link chain in $d \geq 3$ dimensions are equivalent.*

Proof. By Theorem 3.1, in $d \geq 3$ dimensions, each configuration of a closed n-link chain is equivalent to some standard triangular form. Any two standard triangular forms of a closed chain are equivalent via a combination of translation and rotation. □

In $d = 2$ dimensions two standard triangular forms of a closed chain need not be equivalent. The next section discusses this.

4. Configurations in the Plane

This section considers configurations of closed chains in the plane. Clearly, any two standard triangular forms of a closed chain in the plane having the same orientation (clockwise or counterclockwise) are equivalent. Hence, by Theorem 3.1, in the plane, a closed chain has at most two equivalence classes. This section gives a necessary and sufficient condition for a closed chain to have exactly one equivalence class of configurations and gives an $O(n)$ algorithm for moving between any two equivalent configurations.

Lemma 4.1. *Let T be a standard triangular form of a closed chain of n links, let m be the sum of all of the link lengths, and let l_i, l_j, and l_k be the lengths of the first, second, and third longest links of L, respectively. If $l_j + l_k \leq m/2$, then T can be moved to a mirror image using a constant number of simple motions; moreover, these motions can be computed in $O(n)$ time.*

Proof. Let x, y, and z be the joints of the triangle T, labeled as in Definition 3.1, and suppose that $l_j + l_k \leq m/2$. Procedure *reflect_triangle* (Fig. 7) takes T, x, y, and z and moves T to a configuration which is the mirror image of T with respect to the line $M = l(x, z)$. We assume that these joints appear in clockwise order x, y, z on T (the case in which they are in counterclockwise order is analogous).

Let p be the midpoint of the side $\overline{zx}$ and let u and v be the joints (in clockwise order) of the link containing p (if p is a joint, choose $v = p$). See Fig. 8. The algorithm moves the chain to a configuration in which x, y, and z are collinear, then moves y to the other side of M. We now describe how this is accomplished.

Case 1: $|d(x, y) - d(y, z)| \geq |d(x, u) - d(z, u)|$. We can assume that $u \neq z$ in this case; otherwise, because of the inequality, T is degenerate and so is already its own mirror image with respect to M. The algorithm first uses a line-tracking motion with elbows $E(x, y, z)$ and $E(x, u, z)$ to move z toward x along M until y folds. The inequality above ensures that u does not fold before y does. Now the motion of the elbows is reversed but as y begins to unfold, y is moved to the opposite side of M. The motion continues until u straightens (it must straighten before y by the triangle inequality); the chain is now configured as a mirror image of T.

Case 2: $|d(x, y) - d(y, z)| < |d(x, u) - d(z, u)|$. The inequality above guarantees that $v \neq x$. It will be easier to see this, however, once the first (conditional) step of the algorithm is carried out. If $u \neq z$, the algorithm first uses a line-tracking motion with elbows $E(x, y, z)$ and $E(x, u, z)$ to move z toward x along M until u folds. The inequality above ensures that u folds before y does. See Fig. 9. Now it is easy to see that $v \neq x$, since otherwise the links (u, v) and (y, z) would have lengths which sum to more than $m/2$. Because neither of these links is the maximum length link, which lies on the chain from x to y, the second and third longest link lengths would sum to more than $m/2$.

Now, regardless of whether $u = z$, a line-tracking motion with elbows $E(z, y, x)$ and $E(u, v, x)$ is used to move x toward z along M until y folds. We claim that v does not fold before y does, for otherwise links (u, v) and (y, z) have length whose

```
procedure reflect_triangle(T,x,y,z)
{ T is a standard triangular form with joints x, y and z. }

choose M = l(x, z)
if |d(x, y) - d(y, z)| ≥ |d(x, u) - d(z, u)| then
     if u ≠ z then
          move z towards x along M with a line tracking motion
          using elbows E(x, y, z) and E(x, u, z) until y folds
          { The inequality above ensures that u does not fold before y. }

          reverse the preceding motion, moving z away from x
          but moving y to the opposite side of M as y unfolds
          { This moves T to its mirror image with respect to M. }
     else
          stop { T is degenerate and is already its own mirror image. }
     endif
else { |d(x, y) - d(y, z)| < |d(x, u) - d(z, u)| }
{ Claim: v ≠ x. See proof of Lemma 4.1. }
     if u ≠ z, then
          move z toward x along M with a line tracking motion
          using elbows E(x, y, z) and E(x, u, z) until u folds
          { The inequality ensures that u folds before y does. }
          { See Figure 9. }
     endif

     move x toward z along M with a line tracking motion
     using elbows E(z, y, x) and E(u, v, x) until y folds
     { Claim: y folds before v. }

     reverse the preceding motion, moving x away from z
     but moving y to the opposite side of M as y
     begins to unfold.

     if u ≠ z then
          move z away from x along M with a line tracking motion
          using elbows E(x, y, z) and E(x, u, z) until u straightens
          { The resulting configuration is the mirror image of T. }
     endif
endif
end reflect_triangle
```

Fig. 7. The algorithm for moving from standard triangular form to a mirror image when $l_j + l_k \leq m/2$.

sum is greater than $m/2$. However, there is at least one link on the chain from x to y which is at least as long as either of these two links. Therefore, as in the argument that $v \neq x$ above, the lengths of the second and third longest links must also sum to more than $m/2$.

To continue the description of the algorithm, the previous motion of the elbows is reversed, but as y begins to unfold, y is moved to the opposite side of M. The motion continues until v straightens.

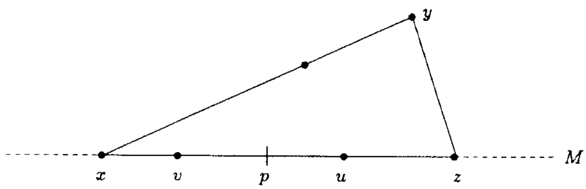

Fig. 8. A closed chain in standard triangular form with the longest link having endpoint x and lying along $\overline{xy}$.

If $u = z$, we have obtained a configuration which is the mirror image of T; otherwise one final line-tracking motion using elbows $E(x, u, z)$ and $E(x, y, z)$ to move z along M away from x is performed until u straightens. □

The preceding lemma can be proved in another way, which, while more direct from a topologist's point of view, does not provide an algorithm for carrying out a reconfiguration physically. The idea of the alternative proof is as follows. It is shown that the set of configurations of a closed chain of given edge lengths is congruent to the set of configurations of that chain with its edges reassembled in any other order. In particular, the number of connected components of the set of configurations is independent of the edge order. This allows convenient assumptions, such as the longest and shortest edges being adjacent, or the second and third longest edges being adjacent, to be made without loss of generality.

The next lemma proves that the condition of the preceding lemma is also a *necessary* condition for inverting a planar configuration, that is, moving a closed chain from a given configuration to a mirror-image configuration. These two lemmas taken together characterize invertibility in terms of the condition on the link lengths. This means that invertibility of closed chains is independent of their initial placement and even of the order of their links.

A sequence (x, y, z) of three noncollinear points in the plane is a *left turn* if it determines a counterclockwise cycle; otherwise the sequence is a *right turn*. The orientation of a sequence of collinear points is undefined.

The key idea behind the proof of the next lemma is the following. Let $L_i = (v_{i-1}, v_i)$ be a link of a closed chain L. Suppose that some configuration of L can be moved to its mirror image with respect to some arbitrary line. Let v_j be any joint of L that is not collinear with L_i in this configuration. Then at some moment during the motion that moves the configuration to its mirror image, v_j

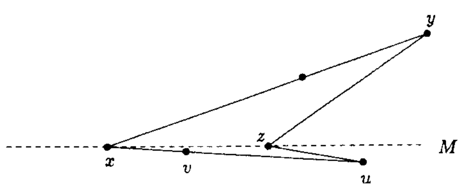

Fig. 9. In moving z toward x, u folds before y.

and L_i must be collinear. This is because the orientation of (v_j, v_{i-1}, v_i) in the original configuration of L differs from its orientation in any mirror image of that configuration. Determining what conditions on the lengths of the links allow each v_j to become collinear with each L_i yields a necessary condition for invertibility.

Lemma 4.2. *Suppose L is a closed chain consisting of $n \geq 3$ links, and suppose the sum of its link lengths is m. A configuration of L can be moved to a mirror-image configuration only if the lengths of the second and third longest links of L sum to no more than m/2.*

Proof. The proof is trivial for $n = 3$. Assume that $n \geq 4$. We show that if the condition is violated, then some joint cannot be positioned on the line determined by some link, which is necessary for invertibility by the key idea. Since the chain has at least four links, by suitably choosing a labeling of the joints we can assume that L_i, L_j, and L_k are the three longest links, that L_j and L_k do not have a joint in common, and that $i < j < k$. See Fig. 10. Note that L_i, L_j, and L_k do not necessarily appear in order of decreasing length and that the configuration of L may not form a simple polygon as in the figure.

If the condition is violated, then

$$l_j + l_k > m/2 \tag{1}$$

$$l_i + l_j > m/2, \tag{2}$$

and

$$l_i + l_k > m/2. \tag{3}$$

To complete the proof, it suffices to show that making v_j and L_i collinear violates one of (1)–(3). Partition L into three chains as follows: the link $L_i = (v_{i-1}, v_i)$, the "left chain" $(v_i, v_{i+1}, \ldots, v_j)$, and the "right chain" $(v_j, v_{j+1}, \ldots, v_{i-1})$. Also, let s_j be the total length of the left chain not including link L_j, and let s_k be the total length of the right chain not including link L_k. Thus $m = l_i + l_j + s_j + l_k + s_k$.

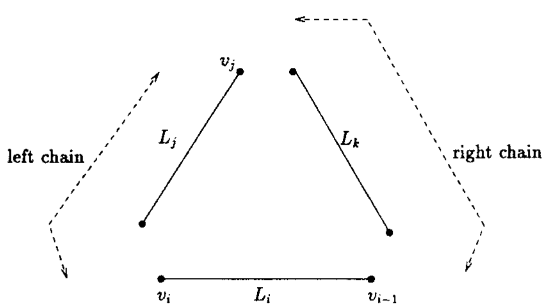

Fig. 10. A polygon with longest links L_i, L_j, and L_k.

For future reference, observe that, regardless of the configuration of the closed chain, the following must hold:

$$d(v_j, v_i) \geq l_j - s_j,$$

$$d(v_j, v_{i-1}) \geq l_k - s_k,$$

$$d(v_j, v_{i-1}) \leq l_k + s_k,$$

and

$$d(v_j, v_i) \leq l_j + s_j.$$

Suppose now that there is a configuration of the chain such that v_j and L_i are collinear. We consider the relative positions of v_{i-1}, v_i, and v_j on the line $l(v_{i-1}, v_i)$.

Case 1: v_j lies between v_i and v_{i-1}. In this case,

$$l_i = d(v_j, v_i) + d(v_j, v_{i-1}) \geq (l_j - s_j) + (l_k - s_k),$$

which implies that

$$m/2 \geq l_j + l_k.$$

This violates inequality (1).

Case 2: v_i lies between v_j and v_{i-1}. In this case,

$$l_k + s_k \geq d(v_j, v_{i-1}) = d(v_j, v_i) + d(v_i, v_{i-1}) \geq (l_j - s_j) + l_i,$$

which implies that

$$m/2 \geq l_j + l_i.$$

This violates inequality (2).

Case 3: v_{i-1} lies between v_j and v_i. In this case,

$$l_j + s_j \geq d(v_j, v_i) = d(v_j, v_{i-1}) + d(v_{i-1}, v_i) \geq (l_k - s_k) + l_i,$$

which implies that

$$m/2 \geq l_k + l_i.$$

This violates inequality (3). □

Theorem 4.1. *A configuration of a closed chain in the plane can be inverted if and only if the lengths of the second and third longest links sum to no more than the sum of the lengths of the remaining links. This can be checked in time proportional to the number of links.*

Proof. Immediate consequence of Lemmas 4.1 and 4.2. □

Theorem 4.2. *Given two configurations of a closed n-link chain in the plane, it can be determined in $O(n)$ time whether the configurations are equivalent. When the two configurations are equivalent, one configuration can be moved to the other with $O(n)$ simple line-tracking motions together with a single rotation and translation of the entire chain.*

Proof. Move each of the two configurations into standard triangular form, which can be done in linear time. Determine, in constant time, whether the triangular forms are mirror images.

If the triangles have the same orientation or are degenerate, move one to the other with a translation or rotation, and undo the appropriate moves to reach the other configuration.

If the triangles are not degenerate and do not have the same orientation, then test in $O(n)$ time whether they are invertible. If they are not invertible, then the original given configurations are not equivalent. If they are invertible, then invert one, translate and rotate it to the other, and undo the appropriate motion to complete the move to the other given configuration. □

5. Conclusion

This paper has considered the reconfiguration of closed, n-link chains in $d \geq 2$ dimensions, where the links are allowed to pass through one another, or in the case of the plane, are allowed to pass over one another. The main algorithmic result of the paper is the following: In dimension $d \geq 2$ every move between equivalent configurations can be accomplished with $O(n)$ simple line-tracking motions (plus a translation and a rotation that does not change joint angles). These motions can be computed in $O(n)$ time on real RAM.

Our algorithmic approach yields the following topological results about the nature of configurations of a closed chain (these results have alternative, more topological proofs):

1. The configurations of a closed n-link chain in $d \geq 3$ dimensions form one equivalence class; that is, every configuration can be moved to every other configuration.
2. The configurations of an n-link chain in $d = 2$ dimensions form one equivalence class if and only if the sum of the lengths of the second and third longest links (which may be equal in length) is at most half the sum of all the lengths. This means that it can be determined in $O(n)$ time on real RAM whether two configurations are equivalent.
3. If the configurations of an n-link chain in $d = 2$ dimensions form more than one equivalence class, then they form exactly two equivalence classes. Each configuration in one class has a mirror image in the other.

It was also pointed out that a set of n lengths can be realized as a configuration

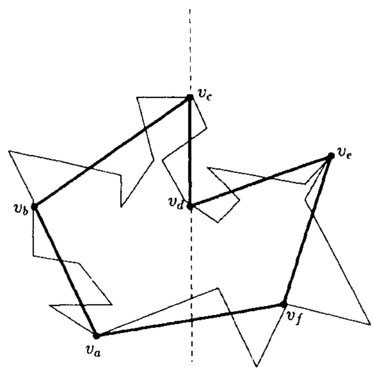

Fig. 11. A generalized line-tracking motion. The thick lines represent "virtual" links; "elbows" $E(v_a, v_b, v_c)$ and $E(v_d, v_e, v_g)$ bend, moving the chain from v_c to v_d along the dashed line. The actual chains (thin lines) that form the virtual links remain rigid throughout the motion.

of a closed chain if and only if no length is greater than the sum of the remaining lengths. Furthermore, if the lengths can be realized as a configuration of a closed chain, then they can in fact be realized as a triangle whose description can be computed in $O(n)$ time.

Another contribution of this paper has been to define the notion of a line-tracking motion. While reconfiguration could be accomplished by other types of motions, this notion has led to algorithms which are not only linear but which are also particularly simple to describe. Consequently, line-tracking motions or some natural generalizations of them could prove useful in other types of reconfiguration problems. While this paper used pairs of elbows cooperating to move a single point along a straight line, it is natural to imagine pairs of "virtual" elbows cooperating to move a point or a section of the linkage along a curve; a virtual elbow would consist of a section of linkage with an internal joint that functions as an elbow joint, while the remaining joint angles remain fixed. (See Fig. 11.)

Among the many open problems that remain, some of the most interesting are the most conceptually simple. What, for example, can be said about the difficulty of determining whether two configurations of a closed chain are equivalent if links are not allowed to cross through or pass over one another? Currently it is still not known whether all configurations of an open chain in the plane are equivalent under this more restrictive type of motion.

Acknowledgment

We thank the anonymous referees for pointing out that configurations of chains have been studied from a topological point of view and for providing references. In particular, we thank one of the referees for providing the sketch mentioned in Section 4 of another proof method for the topological results.

References

1. J. Canny. *Complexity of Robotic Motion Planning.* (ACM Doctoral Dissertation Award, 1987). MIT Press, Cambridge, MA, 1988.
2. G. M. Crippen and T. F. Havel. *Distance Geometry and Molecular Conformation.* Wiley, New York, 1988.
3. J.-C. Hausmann. Sur la topologie des bras articulés. In S. Jackowski, B. Oliver, and K. Pawałowski, editors, *Algebraic Topology, Poznań*, 1989, pp. 146–159 (Proceedings of a conference held in Poznań, Poland, June 22–27, 1989.) Lecture Notes in Mathematics, vol. 1474, Springer-Verlag, Berlin, 1991.
4. J. Hopcroft, D. Joseph, and S. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. Comput.*, 13:610–629, 1984.
5. J. Hopcroft, D. Joseph, and S. Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Comput.*, 14:315–333, 1985.
6. B. Jaggi. Punktmengen mit Vorgeschriebenen Distanzen und Ihre Konfigurationsraüme. Inauguraldissertation, Bern, 1992.
7. D. Joseph and W. H. Plantinga. On the complexity of reachability and motion planning questions. *Proceedings of the Symposium on Computational Geometry*, pp. 62–66, June 1985.
8. V. Kantabutra. Motions of a short-linked robot arm in a square. *Discrete Comput. Geom.*, 7:69–76, 1992.
9. V. Kantabutra and S. R. Kosaraju. New algorithms for multilink robot arms. *J. Comput. System Sci.*, 32:136–153, 1986.
10. W. Lenhart and S. Whitesides. Turning a polygon inside-out. *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pp. 66–69, August 1991.
11. W. Lenhart and S. Whitesides. Reconfiguration with line tracking motions. *Proceedings of the 4th Canadian Conference on Computational Geometry*, pp. 198–203, August 1992.
12. J. Reif. Complexity of the mover's problem and generalizations. *Proceedings of the 20th Symposium on the Foundations of Computer Science*, pp. 421–427, 1979.
13. J. Schwartz and M. Sharir. On the "piano mover's" problem, II. General techniques for computing topological properties of real algebraic manifolds. *Adv. in Appl. Math.*, 4:298–351, 1983.
14. J. Schwartz and C. Yap, editors. *Algorithmic and Geometric Robotics.* Erlbaum, Hillsdale, NJ, 1987.
15. J. Schwartz, C. Yap, and J. Hopcroft, editors. *Planning, Geometry and Complexity of Robot Motion.* Ablex, Norwood, NJ, 1987.
16. S. Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Austral. Comput. J.*, Special Issue on Algorithms, 24(2):42–50, 1992.
17. S. Whitesides and R. Zhao. On the placement of euclidean trees. Technical Report TR-SOCS-89.03. School of Computer Science, McGill University, 1989.

Received November 30, 1992, and in revised form September 13, 1993, and June 1, 1994.