Reconfigurations of Polygonal Structures

Greg Aloupis

School of Computer Science
McGill University
Montreal, Canada

January 2005

A thesis submitted to McGill University
in partial fulfillment of the requirements for
the degree of Doctorate of Philosophy

Abstract

This thesis contains new results on the subject of polygonal structure reconfiguration. Specifically, the types of structures considered here are polygons, polygonal chains, triangulations, and polyhedral surfaces. A sequence of vertices (points), successively joined by straight edges, is a *polygonal chain*. If the sequence is cyclic, then the object is a *polygon*. A planar triangulation is a set of vertices with a maximal number of non-crossing straight edges joining them. A polyhedral surface is a three-dimensional structure consisting of flat polygonal faces that are joined by common edges.

For each of these structures there exist several methods of reconfiguration. Any such method must provide a well-defined way of transforming one instance of a structure to any other. Several types of reconfigurations are reviewed in the introduction, which is followed by new results. We begin with efficient algorithms for comparing monotone chains. Next, we prove that flat chains with unit-length edges and angles within a wide range always admit reconfigurations, under the dihedral model of motion. In this model, angles and edge lengths are preserved. For the universal model, where only edge lengths are preserved, several types of hexagons that cannot be reconfigured are exhibited. New bounds are provided for the number of operations required to reconfigure between triangulations, using “point moves” and “edge flips”. Finally, unfolding motions are proven to always be possible for specific slices of polyhedra.

Résumé

Cette thèse présente de nouveaux résultats dans le domaine de la reconfiguration de structures polygonales. En particulier, les polygones, chaînes polygonales, triangulations, et surfaces polyédrales seront considérées ici. Une séquence des points (sommets) successivement connectés par des segments de droites (arêtes), est une *chaîne polygonale*. Si la séquence est cyclique, l'objet est un *polygone*. Une triangulation planaire est un ensemble de sommets connectés par un nombre maximal d'arêtes dont les intérieurs sont mutuellement disjoints. Une surface polyédrale est une structure tridimensionnelle composée de faces polygonales planaires connectées par des arêtes communes.

Pour chacune de ces structures, plusieurs méthodes de reconfiguration existent. Une méthode spécifie comment transformer une instance d'une structure en une autre. Un survol des types de reconfigurations est présenté dans l'introduction, et les chapitres suivants prsentent de nouveaux résultats. Il est démontré dans cette thèse que les chaînes plates avec des arétes de longeur unitaire et certaines contraintes sur les angles admettent des reconfigurations dans le modèle diédral. Dans ce modèle, des angles et les longueurs d'arête sont préservés. Dans le modèle universel, pur lequel seules les longueurs d'arête sont préservées, plusieurs types d'hexagones qui ne peuvent pas être reconfigurés sont preséntés. De nouvelles bornes sont prouvées sur le nombre d'opérations nécessaires pour transformer une triangulation à une autre en utilisant les *mouvements des points* et les *basculements d'arêtes*. Finalement, il est montré comment certaines tranches de polyèdre peuvent toujours être dépliées.

Statement of Originality

Except for results whose authors have been cited where first introduced, all elements of this thesis are original contributions to knowledge. Assistance has been received only where mentioned in the acknowledgements.

Acknowledgements

The results in Chapter 3 have been submitted to the *Computer Music Journal.* My coauthors are Thomas Fevens, Stefan Langerman, Tomomi Matsui, Antonio Mesa, Yurai Nuñez, David Rappaport and Godfried Toussaint. We are grateful to Godfried for suggesting the main problem to us. Preliminary results appeared at the 15th Canadian Conference on Computational Geometry [AFL$^+$03].

The results in Chapter 4 have been submitted to *Algorithmica* upon invitation by the program committee of the 12th International Symposium on Graph Drawing, where preliminary results appeared [ABM04]. My coauthors are Prosenjit Bose and Pat Morin. I thank them for inviting me to visit Carleton University to work on reconfiguring triangulations.

The results in Chapter 5 are published in *Contributions to Algebra and Geometry* [AET04]. My coauthors are Günter Ewald and Godfried Toussaint.

The results in Chapter 6 appear only here. They are an extension of work that appeared at the 14th Canadian Conference on Computational Geometry [ADM$^+$02]. The coauthors of that paper are Erik Demaine, Henk Meijer, Joseph O'Rourke, Ileana Streinu and Godfried Toussaint.

The first two sections of Chapter 7 have been submitted to *Computational Geometry: Theory and Applications* upon invitation by the program committee of the 16th Canadian Conference on Computational Geometry, where preliminary results appeared [ADL$^+$04]. My coauthors are Erik Demaine, Stefan Langerman, Pat Morin,

Joseph O'Rourke, Ileana Steinu and Godfried Toussaint. All remaining sections of that chapter appear only here.

I thank David, Erik, Godfried, Günter, Henk, Ileana, Jit, Joe, Pat, Stefan, Tom, Tomomi, Tony, and Yurai for our exciting collaboration.

All portions of this thesis that have been published elsewhere were originally written by myself and carefully corrected by coauthors. Proofs and algorithms included here were designed either on my own or through discussions with coauthors. In each case, I feel that my contribution was essential. For this reason, I screened out the work that appears in [ABD$^+$02] and [ADD$^+$02]. These two papers include my first work on reconfigurations and are mentioned in Chapters 2 and 6, but in my opinion my contributions were not significant enough to justify inclusion in a doctorate thesis.

Financial assistance for two years of this research was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). For the remaining time, I received an equally generous stipend from my supervisor, Godfried Toussaint.

I am grateful to Godfried for many reasons beyond that of his support throughout my graduate studies. He has a unique ability of discovering problems and new fields of research in geometry, and is always enthusiastic in sharing these discoveries with colleagues and students. The topic of reconfigurations was just one of several which he suggested to me as I began my degree. Godfried also made sure that I met and collaborated with many computational geometry experts, by sending me to several conferences, insisting that I give talks, and organizing workshops on the topics that I was studying. It is no coincidence that most of my work was initiated away from home. For example, the work on dihedral motions was initiated in February of 2002, at an annual winter workshop organized by Godfried in Barbados. The work on unfolding polyhedral bands started at the workshop of 2004. The problem of computing the similarity of chains was offered by Godfried at a workshop in Havana, in April of 2003. Workshops in Barbados and Havana are a reflection of Godfried's philosophy on life. Life should be enjoyed, and there is no exception when it comes to research. Even though my hardest working days were during my one week visits to the tropics, at the same time they were the most relaxing, and I always couldn't wait to return. In addition to inventing the great tropical incentive for research, Godfried has the patience and ability to understand how to work with everyone. Those who have met me know that my schedule is unusual, both on a daily and on a yearly level. Fortunately, I was given the maximum possible amount of freedom to work whenever and wherever I preferred. Overall, I would say that I was lucky to find a supervisor who treated me with respect as an equal and a friend.

I would also like to thank the rest of the professors (David Avis, Luc Devroye, Bruce Reed, Sue Whitesides) in the Computational Geometry group at McGill for creating perhaps the most enjoyable academic environment in the world. In particular, David and Luc were on my thesis committee and I appreciate the advice that they gave me.

I decided to skip mentioning my friends, since I can always do that in person, and I would probably forget some of them... not to mention that they will probably never read this. Though there is one thing that I must say: yes, my thesis is about lines and dots.

I thank my family for all the love, encouragement and interest in my career, even though they sometimes get some details wrong. Flips and flipturns are not stretching routines, the self-avoiding walk is not a term from psychology, and most of all, I am not a Biospherical Engineer.

This thesis is dedicated to my parents, who insisted that I leave Greece for Canada at the age of 17 to obtain a better university education, even though that meant that I would be away for ten years.

Contents

List of Figures

Chapter 1

Introduction

Consider the puzzle described here and illustrated in Figure 1.1: You are given two drawings of the same polygonal chain, A and B. This means that both A and B have the same sequence of fixed edge lengths. In both drawings no edge crosses over another. You are free to move A around and change any angle between neighbouring edges, as long as no crossings are introduced. Can you move A into the same configuration as B?

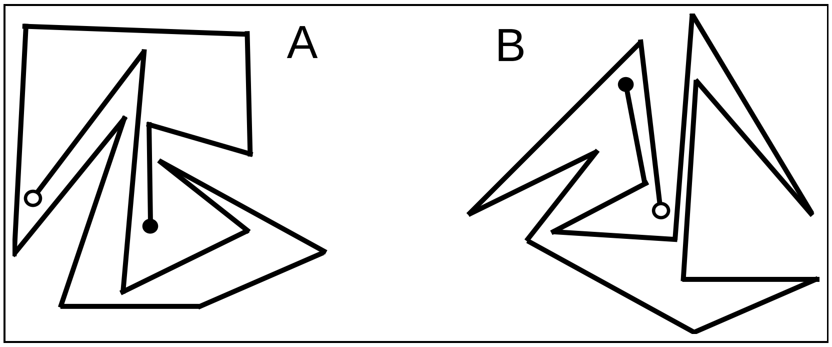

Figure 1.1: A puzzle: can you move A to B without crossing edges?

This thesis is a collection of problems such as the above, with their solutions. Simply put, the main problem considered here is this. Given a polygonal object and a set of operations that reconfigure it, can the object be transformed to a new specified configuration?

The term “polygonal object” is used to categorize several structures. The characteristic that makes objects polygonal is the presence of certain common elements such as vertices and straight edges. Precise definitions of each structure to be considered are deferred for now.

Most operations that we[1] consider produce only local changes to objects, which means that to reconfigure between two arbitrary polygonal objects we may need a sequence of operations. Regardless of the set of reconfiguring operations that is selected, we can ask the same types of questions: Can a given object be reconfigured into a specified “target” configuration? Can we prove that all configurations of an object are reachable from one another? What can we say about the time complexity of answering these questions for a specific given object, and what about algorithms that calculate the necessary motions?

Motivation

Why study polygonal reconfigurations at all? The reason that first comes to mind is for pure mathematical beauty. Any reader who has ever been interested in solving a mathematical puzzle will most likely spend more than just a few moments contemplating the general problems described in the following chapters. The simple geometric nature of these problems makes them accessible to nearly everyone. At the very least, the problem formulations ought to be clear and intuitive. After all, most objects that we encounter in life can be abstracted or approximated by lines and points. And, the objects that tend to catch our attention are those that move about, or in other words, reconfigure.

Having nearly included the entire class of moving objects on the face of this earth within the scope of this thesis, perhaps it is appropriate to mention a few examples

[1] By “we” I usually mean my coauthors and myself.

that are more specific. Where do we really find objects made of lines and dots that change shape? A quick answer is, everywhere in computer simulations that model our kinetic world. It is often difficult or impossible to manipulate exact descriptions of real-life objects. When using the discrete-minded computer, a standard method is to approximate reality with polygonal chains, triangular meshes and so on. For example, it is common knowledge among video-gamers that the racing cars, space-ships and alien invaders are typically displayed and represented as polyhedral surfaces. The illusion becomes increasingly better as the number of points (or polyhedral faces) used in the approximation is increased. So, a fancy car really ends up being a surface made of a few thousand triangles. If you expect this car not to look unscratched after a severe collision, then you are expecting it to be reconfigured.

Moving on to more noble areas of motivation than computer games, we can imagine that a polyhedral car (Fig. 1.2) might also be found in simulations that have to do with design or safety. But to change areas entirely, consider the field of molecular chemistry. Polymers and DNA are typical molecules that have long backbone structures. The image of atoms (points) joined by chemical bonds (edges) leaps to mind (Fig. 1.3). Many studies in this field involve reconfigurations of molecular shape and position.

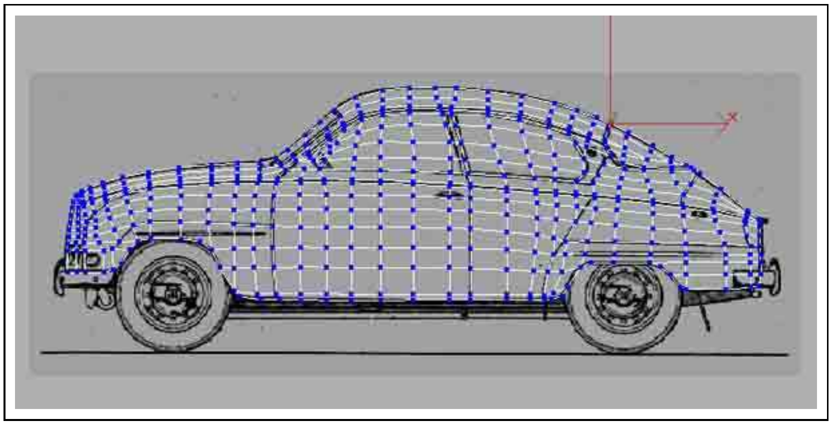

Figure 1.2: A polygonal mesh covering the blueprint of a car.
(courtesy of Lars “GRID” Martensson from www.3dluvr.com/grid)

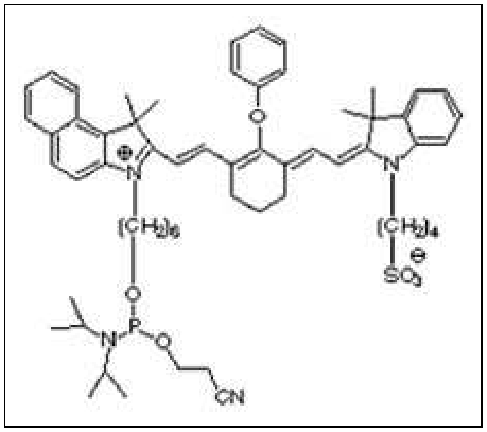

Figure 1.3: Fluorochrome polymer.
(from http://microgravity.grc.nasa.gov/grcbio/SRF.html)

Or what about the exciting world of robots? If you have ever witnessed the use of robotic arms in automated factory productions, no doubt you will have been impressed at how fast and accurately several multi-linked arms must coordinate. Whether these robotic arms are represented as simple chains or as polyhedral surfaces, collision-avoidance is a natural issue to study geometrically.

Further on in this thesis, you will see that links have been made between polygonal reconfigurations and even more surprising topics, such as musical comparison and retrieval, manufacturing sheet metal, counting random walks, and even locating a river shore on a foggy night!

Interesting links may be found everywhere, but we should maintain a bit of realism. Even though the world may be abstracted by polygonal surfaces, this does not directly imply that all problems can be easily modeled and solved via this approximation. Mathematicians and computer scientists are still tormented by the simplest problems (in appearance, but not in solution). The results presented in this thesis are typically about special cases of well-known open problems in the field of computational geometry, but no patent applications will be filled out, and it is unlikely that these theorems will be used by major game-developing, pharmaceutical or robotics companies any time soon. However, as has been proven on several occasions, such problems often end up finding direct purpose in the most unlikely places.

The computational model used

The primary concern in many of the problems studied here is whether a reconfiguration can be completed, and as such the only necessary background is a sense of "geometric space". On the other hand, in some applications, time and storage space are essential, and for these we must define a model of computation.

When describing the asymptotic computational complexity of algorithms and problems with respect to the size of input n, the "universally accepted" notation of Knuth will be used (see [Knu76, PS85]). Thus for some function f_n and for large enough n, we say that $O(f_n)$ is at most a constant multiple of f_n, $\Omega(f_n)$ is at least a constant multiple of f_n, and $\Theta(f_n)$ is both $O(f_n)$ and $\Omega(f_n)$. Upper bounds for the time and space used by algorithms will be in the *real RAM* model of computation (see [PS85]). According to this model, only arithmetic operations $(+, -, \times, /)$ and comparisons are allowed for real numbers of infinite precision. Occasionally this is extended to include certain functions such as trigonometric, exponential, k-th root, etc. Each arithmetic operation costs one unit of time. The same applies for storing and retrieving each real number, where the storage of a number also requires a unit of space. Lower bounds for the worst case complexity of algorithms *or* problems are in the *algebraic decision tree* model (see [PS85]). A (binary) algebraic decision tree represents the set of possible sequences of operations made by a RAM algorithm, where the root of the tree represents the first operation performed and each leaf represents a possible output. Any path from the root to a leaf in the tree also represents the time for a specific sequence of operations. The worst-case performance of an algorithm is specified by the longest path from the root to some leaf. Obtaining lower bounds for the time complexity of problems essentially involves determining the path-length size that an algebraic decision tree must have for the tree to be able to produce the desired output for any possible input. For example, it can be shown that any algebraic decision tree which is capable of sorting n real numbers based on comparisons alone must have a path with length $\Omega(n \log n)$, and therefore *any* RAM algorithm restricted to comparisons must take $\Omega(n \log n)$ time *in the worst case* to sort n real numbers (see [CLRS01]). For a discussion on the connection between the RAM and algebraic decision tree models, see [PS80].

According to the above, if an algorithm takes $O(f_n)$ time to solve a problem and there is an instance of the problem known to require $\Omega(f_n)$ time, then the algorithm is worst-case optimal. If exactly k operations are *always sufficient* to perform a task, then k is an upper bound. If this is *sometimes necessary*, i.e. if there also exists an example/instance where k operations are required, then we say that the upper bound is tight.

In general, algorithms are considered "efficient" if their run-time can be upper-bounded by a polynomial expression in terms of n, in the asymptotic notation given above. Of course, efficiency is a relative concept. In most cases we are not exactly happy with just any polynomial run-time.

Finally, the class of *NP-hard* problems should be mentioned, since these turn up at various points later on. Let the input size of a problem be the number of bits needed to encode it. For example, a vertex with some extremely large coordinate position is not just an input of size one (for rational numbers, a logarithmic number of bits suffices though). Consider the class of problems that can be solved in polynomial time by a deterministic algorithm (class P). This is a subset of a broader class which also includes problems whose solutions can be verified in polynomial time but for which no polynomial algorithm is known (class NP[2]). Informally, a problem is NP-hard if a deterministic polynomial algorithm for it would also imply the existence of such

[2] NP stands for "non-deterministic polynomial" since a solution could be produced and verified in polynomial time by a non-deterministic algorithm that guesses the correct solution.

an algorithm for all problems in NP. In other words, every instance of a problem in NP can be transformed in polynomial time to an NP-hard problem. There are thousands of NP-hard problems, and by now it seems unlikely that anybody will find a polynomial-time algorithm to solve one of them.

PSPACE-hard problems are even more impractical, since it is only known that they can be solved with a polynomial amount of space, using an unlimited amount of time. No more detail on these matters is given here, since mention of these classes is mainly restricted to background material of this thesis. A standard reference is the text by Garey and Johnson [GJ79].

Elementary geometric objects and notation

In general, definitions are given when needed throughout the text. Only a few concepts are mentioned here, since they are used repeatedly in this thesis.

The symbol $\mathbb{R}^d$ denotes d-dimensional space. Two consecutive symbols in the form xy represent either a segment from point x to point y, or a line that passes through both points (though edges and lines are sometimes denoted by a single symbol when the vertices are not important). The case will be clear from context. When applicable, it is understood that the "direction" of a segment or line is from x to y. This allows us to clearly define when a point is to the left or right of a segment. The point p is to the left of the segment xy if we make a left turn moving from x to y and then to p.

A polygonal chain is a sequence of edges, joined consecutively at endpoints, as the one shown in Figure 1.1. A cyclic chain is a *polygon*. In $\mathbb{R}^2$, polygons divide the plane into an interior and an exterior region. It is common to consider the interior region as part of the polygon, but this will not be done here unless specifically mentioned or implied by context. We say that a polygon or chain is simple if no edges intersect, with the obvious exception of common endpoints for neighbouring edges.

A polygon is convex if any two points on its boundary can be connected by a segment through the interior that does not cross any polygon edges. If we assign a clockwise direction to the boundary of a convex polygon, then all turns are to the right. If the polygon is non-convex, any left turn occurs at a *reflex* vertex. The *convex hull* of a set of points in the plane is the minimum-area convex polygon enclosing the points. Informally, imagine shrinking a rubber band around the points, or running around them as fast as possible. The convex hull of a polygon can be obtained visually by considering only its vertices. A *pocket* of a simple non-convex polygon is a subchain that is strictly in the interior of the convex hull, except for its endpoints that are hull vertices. The *lid* of a pocket is the hull segment between the two endpoints. Note that a pocket is often defined as the polygon enclosed by the chain and lid mentioned in the preceding. These concepts are illustrated in Figure 1.4.

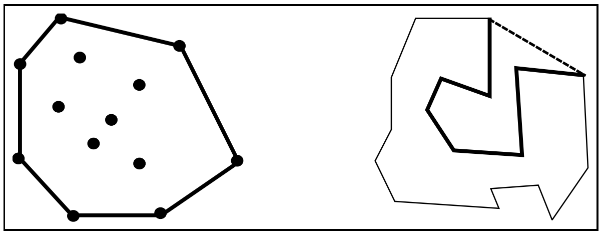

Figure 1.4: Left: A convex polygon that is the hull of a point set. Right: a non-convex polygon with highlighted pocket and lid.

A chain is *monotone* with respect to a direction if any line orthogonal to that direction intersects the chain at most once. A monotone polygon can be decomposed into two monotone chains. A polygon is *star-shaped* if there exists an interior region (the *kernel*) from which every boundary point is visible. In other words a segment from the kernel to any boundary point does not intersect the boundary (see Figure 1.5).

Polygons and chains are just special instances of geometric graphs. A geometric graph is a set of non-crossing edges embedded on a set of vertices. A tree is a connected graph with no cycles. The triangulation of a given point set is a graph with a maximum number of edges inserted between the points (see Figure 1.6).

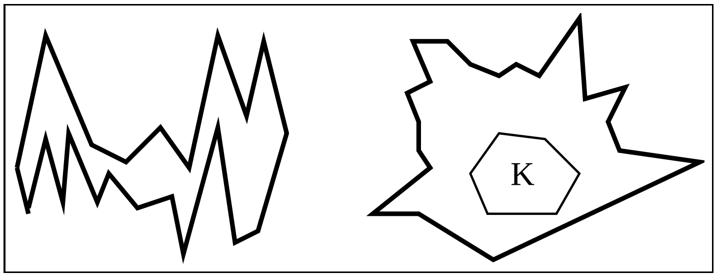

Figure 1.5: Left: a monotone polygon. Right: A star-shaped polygon. The kernel K is also shown.

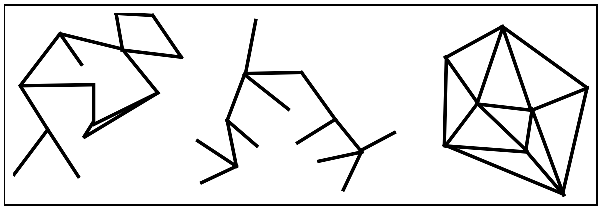

Figure 1.6: From left to right: a geometric graph; a tree; a triangulation.

Configuration space and canonical forms

The *configuration space* of an object is simply the set of valid configurations into which it can be positioned. For example, a planar chain's configuration could be specified by the (x, y) coordinates of each vertex. Any given configuration of a two-edge chain is a point in 6-dimensional space (with coordinates $x_1, y_1, x_2, y_2, x_3, y_3$). It is easy to see that not all 6-tuples of such coordinates correspond to a valid configuration of a chain with specified edge lengths. Reconfiguring between two configurations of an object is equivalent to moving along some path from one point to another in configuration space. Any two valid points in this space, that are connected by a valid path, exist in one *component* of space. In this case we say that the represented configurations are *connected*.

A brief note on the importance of *canonical configurations* is now given. A canonical configuration (or canonical form) is an intermediate between the source and target objects of a reconfiguration. Typically it has a very simple shape. For example, straight chains or convex polygons often serve as canonical forms. Once such a canonical form is adopted, a solution to a reconfiguration problem can be obtained by proving that both the source and target can be reconfigured to canonical form. One of these motions can be inverted to complete the original reconfiguration. Think of having to prove that a route exists between any two locations that are next to a road. This can be done by proving that all roads lead to Rome.

In general, if an object cannot reach a desired canonical configuration, we say that it is *locked* (e.g. a chain that cannot be straightened).

Contributions of this thesis

The main contributions of this thesis are detailed in Chapters (3-7). The polygonal objects considered in Chapter 3 are planar monotone chains. They are moved only with translations, so that any two of them can be compared by superimposing them as best as possible. In Chapter 4, planar triangulations are considered. The question is whether any two triangulations are connected by reconfigurations consisting of two types of operations: *edge flips* and *point moves*. In Chapter 5, polygons in $\mathbb{R}^3$ are the object of interest, and the same general question is asked. This time the *universal* model of motion is considered, which is one of the least restrictive methods of reconfiguration. In this model, the two edges at any vertex are allowed to move to any relative position, as long as no edge intersections occur and edge lengths remain fixed. In Chapter 6 we continue with chains and *dihedral motions*. These motions are exactly what is allowed in the standard chemistry "ball-and-stick" tool-kit: edge lengths and angles between edges must remain fixed. Finally, Chapter 7 deals with *unfolding motions* for polyhedral surfaces. Here, the edges between polyhedral faces behave as hinges and each face must remain rigid.

How to read the rest of this thesis

Chapter 2 contains a review of several types of reconfigurations, including those mentioned above. It is recommended (but not necessary for a reader familiar with the reconfiguration models) to read this before proceeding to the following chapters (3-7) which focus on new results. These chapters may be read in any order, as they are self-contained and include necessary definitions and essential references.

Chapter 2

Review of Polygonal Reconfigurations

This chapter contains a review of several types of reconfigurations. Most of the focus is placed on those reconfigurations that are relevant to the new contributions of this thesis. It should be mentioned that, recently, extensive survey papers on linkages, folding and unfolding have been published (e.g. [ST02, Tou99a, Tou99b, DOar]). Also recent are two doctoral theses that are considered masterpieces, at least by those in the field [Sos01, Dem01]. Therefore including an exhaustive survey here not only serves little purpose, but would make it impossible to avoid copying the work of others. The references provided here are selected highlights, and/or work that I personally found interesting. Unavoidably, there is a large amount of overlap with the papers mentioned in the preceding. For the sake of keeping the bibliography uncluttered, only journal publications are referenced whenever possible. These tend to be more accessible and detailed than any preliminary versions which may have appeared at conference proceedings or as technical reports.

2.1 Triangulations

2.1.1 Concepts and algorithms

A geometric *near-triangulation* consists of a set of points, pairs of which are joined by straight edges, so that no two edges intersect at their interiors and no more edges can be added. In other words it is a maximal plane geometric graph. As a result, the convex hull of the point set is subdivided into triangular faces. The region outside of the hull is known as the *outer face.* If the outer face is incident to only three vertices (i.e. the hull is defined by three vertices), then the object is plainly called a *triangulation.* In the literature this term is often used to include near-triangulations as well. This will be done here too, though in Chapter 4 the distinction will be necessary.

An *edge flip* on a triangulation is simply the deletion of an edge, followed by the insertion of another edge such that the resulting graph remains a triangulation (see Figure 2.1). This can only be done when the edge to be deleted is the base of two triangles whose union is a convex quadrilateral.

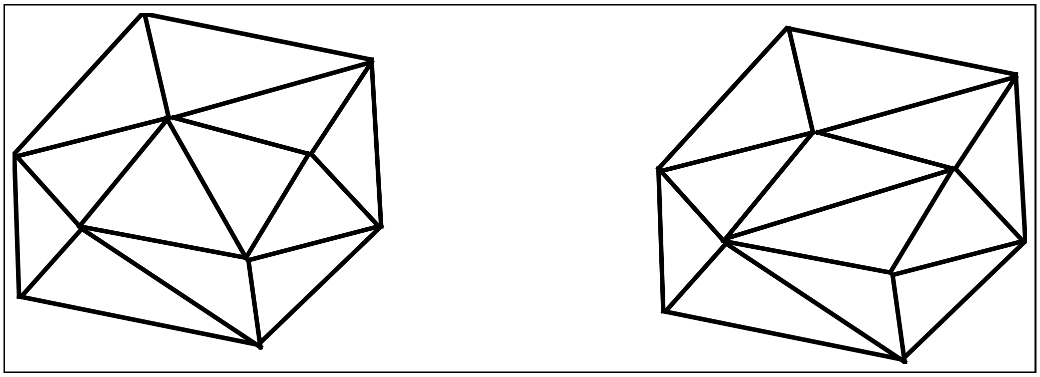

Figure 2.1: An edge flip.

The origins of the edge flip appear to be in the work of Wagner [Wag36], who was investigating the four-colour theorem for planar graphs. However, this was done in the combinatorial setting (meaning that edges are not required to be straight). Wagner proved that a finite sequence of combinatorial edge flips always suffices to reconfigure a triangulation on n points so that it becomes isomorphic to any other triangulation on n possibly different points. Komuro [Kom97] showed that in fact $O(n)$ edge flips suffice. Gao, Urrutia and Wang [GUW01] showed that in the case of labelled triangulations $O(n \log n)$ flips suffice. Bose et al. [BCG$^+$03] showed that $O(\log n)$ *simultaneous* edge flips suffice and are sometimes necessary. Dewdney [Dew73] appears to be the first to have extended this research to other surfaces, by proving Wagner's results on torus graphs. See two recent papers by Cortes et al. [CGMN02, CGH$^+$03] for complete coverage of flips on non-planar surfaces.

Notice that there are valid edge flips (and triangulations) in the combinatorial setting that are no longer valid in the geometric setting, as can be seen in Figure 2.2. Thus the results mentioned directly above do not necessarily hold in the geometric setting.

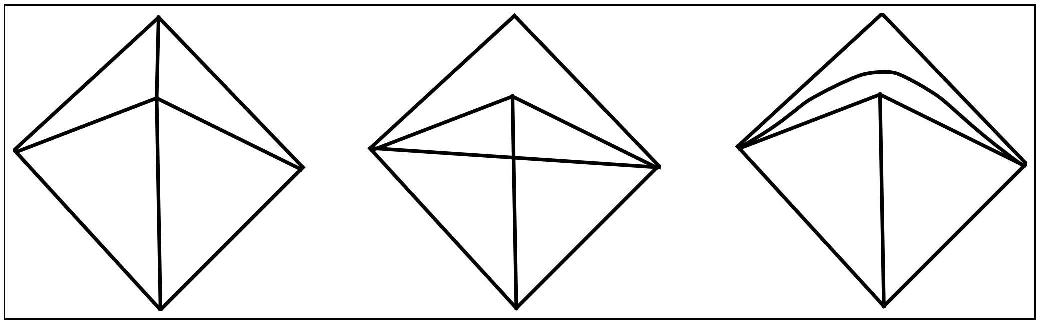

Figure 2.2: Left: initial triangulation. Middle: invalid geometric flip. Right: valid combinatorial flip.

Lawson [Law72] proved that a finite number of edge flips suffices to reconfigure between geometric triangulations *on the same point set.* In a subsequent paper that is often cited as the first proof of this theorem [Law77], Lawson showed that any triangulation can be converted to the Delaunay triangulation by special flips that locally improve Delaunay constraints[1]. This is done for the purpose of obtaining nice $3D$ surface interpolations. Clearly the Delaunay triangulation serves as a canonical configuration, and thus the theorem is implied again.

Fortune [For93] showed that $O(n^2)$ special edge flips suffice to reconfigure a triangulation to the Delaunay triangulation and that the bound is tight. For general edge flips there was still hope for a sub-quadratic bound until Hurtado, Noy and Urrutia [HNU99] showed that $\Omega(n^2)$ flips are sometimes necessary. They also proved that $O(kn)$ flips suffice, where k is the number of convex layers in the point set. Another interesting result of theirs is that $O(n + r^2)$ flips suffice to reconfigure a triangulated polygon with n vertices, r of which are reflex. In addition it is shown that roughly $(n - 4)/2$ edges can be flipped at any time.

Hanke, Ottman and Schuierer [HOS96] showed that the number of intersections between two triangulations is an upper bound on the number of edge flips needed to reconfigure between them. The authors also note that extremely little is known about the complexity of computing the *exact* number of edge flips necessary for reconfiguration between two specific triangulations.

Galtier et al. [GHN$^+$03] showed that $O(n)$ simultaneous edge flips are always sufficient and sometimes necessary. Their results also hold for triangulated polygons. It is interesting that $(n - 4)/6$ edges can be flipped in parallel.

Note the discrepancy between the combinatorial and the geometric settings, which was implied earlier. In the combinatorial setting, from a given triangulation on n points, it is possible to reach *every* triangulation that can be formed on *any* n vertices (isomorphically, as mentioned). So, essentially it is possible to enumerate all different combinatorial triangulations that can be formed with n points, using the flip operation. On the other hand, in the geometric setting it will not be possible to embed all n-point triangulations on a given point set. For example, in the point set

[1] In the Delaunay triangulation the circumscribing circle of any triangle contains no points.

shown on the left in Figure 2.3, the white vertices are connected as a K_4 (complete graph on 4 vertices). This is not possible on the point set on the right.

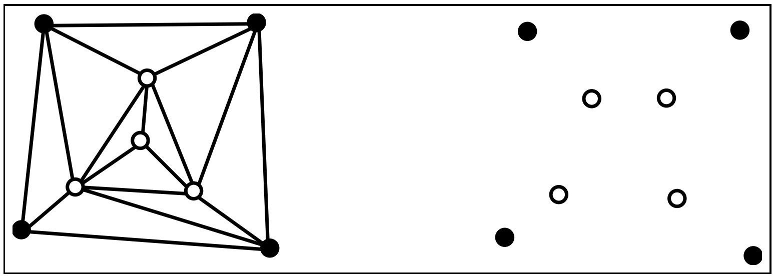

Figure 2.3: Discrepancy between combinatorial and geometric setting.

It is this discrepancy that sparked the work of Abellanas et al. [ABG$^+$04], who introduced the *point move* in the context of reconfiguring triangulations. A point move is the modification of the coordinates of one vertex such that, after the modification, the graph remains a geometric triangulation (see Figure 2.4).

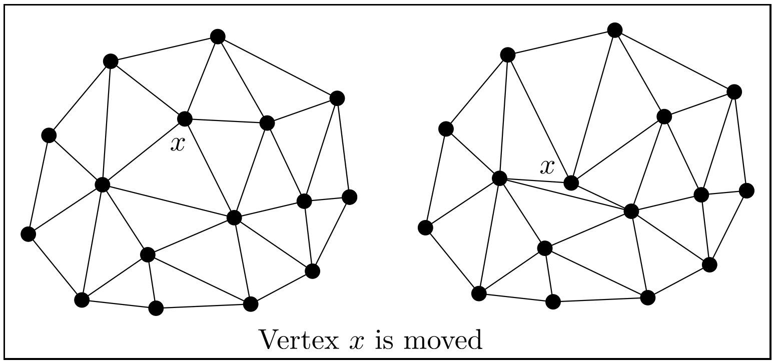

Figure 2.4: Illustration of a valid point move.

Abellanas et al. showed that with $O(n^2)$ edge flips and $O(n)$ point moves, any geometric triangulation on n points can be transformed to any other geometric triangulation on n possibly different points. However, the number of convex hull points must be equal because it determines the number of edges in a triangulation. For triangulations with labelled points, a quadratic number of flips and moves are used.

It should be mentioned that in this work all points were restricted to be positioned on a $5n$ x $5n$ grid. These results can be viewed as a “geometric equivalent” to Wagner’s result. In Chapter 4 we show that the total number of operations can be reduced to $O(n \log n)$ even for labelled triangulations, if the points are not restricted to be on a grid.

2.1.2 Motivation and further comments

In several applications, such as terrain data interpolation, a triangulation is sought which has an optimum quality with respect to some parameters (as was the case in Lawson’s work). Often there is no direct method to compute the optimum triangulation, and standard techniques such as iterative search or simulated annealing are used to find it. These techniques rely on a method for local improvement, such as the edge flip [AKTvD00, DLR90, BHYJ88].

As with many geometric structures, there is much interest in enumerating triangulations. The total number of triangulations is only known for all n up to 20 or so. The edge flip is useful for generating new triangulations as a local search function in exhaustive search algorithms, such as the reverse-search of Avis and Fukuda [AF96]. For convex point sets, there is an isomorphism between the edge flip and the rotation operation on binary trees. For more information on this topic, which is immediately related to enumerating triangulations, see [HN99].

In graphics applications, one problem is how to *morph* polygons or triangulations. Floater and Gotsman [FG99] (see also [SG00]), study how to do this by moving vertices on non-linear continuous paths. This could be approximated by piecewise-linear paths generated by point moves. As the authors mention, even edge flips have been used for this task.

An interesting conjecture, supported by very convincing experimental results, is that any two planar n-point sets with the same number of hull points can be triangulated in a topologically equivalent manner [AAHK03][2]. Thus, if point moves are undesirable (but unavoidable), for triangulations on different point sets it would be possible to use flips on each to reach the common topological form, and then try to use a small constant number of moves per point to complete the reconfiguration.

A different technique of reconfiguring triangulations is implicit in the work of El-Gindy and Toussaint [ET85], who gave efficient algorithms to insert and delete edges. This is done by altogether reconstructing the neighbourhood of an inserted/deleted edge, which means that many edges might be affected in one move. Reconfiguration between two arbitrary triangulations has not been studied in this context.

The generalizations (and difficulties) of flipping in higher dimensions are discussed in [ES96, AAH02a] and references therein.

Finally, flip operations have also been defined on *pseudo-triangulations*. A pseudo-triangle is a polygon with exactly three convex vertices and an arbitrary number of reflex vertices in between. These objects have found many uses in computational geometry, rigidity theory and motion planning (such as polygon convexification, covered in section 2.3.2). These issues are discussed in [Str04].

2.2 Flips and flipturns

2.2.1 Flips

In 1935, Erdős [Erd35] proposed the following reconfiguration for simple planar polygons: perform a simultaneous reflection of all pockets about their corresponding lids, and repeat this for each new polygon obtained in this way. Erdős conjectured that a convex polygon will be obtained after a finite number of iterations. In 1939, Nagy [Nag39] demonstrated that simultaneous reflections may lead to a non-simple polygon, thus creating an undefined situation, since pockets are only defined on sim-

[2] also see this paper for more references to morphing.

ple polygons. Instead, he modified the problem to reflect one pocket at a time, and proved the conjecture. Each step is called a *flip* (see Figure 2.5).

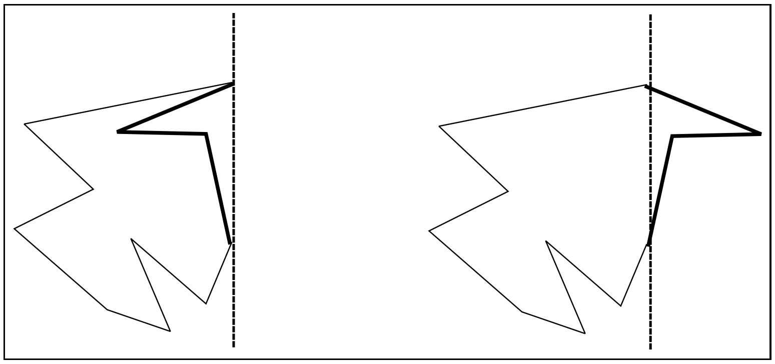

Figure 2.5: A flip: one pocket is reflected about its lid.

Since the solution by Nagy, this problem has been rediscovered and resolved several times (at least once every decade!). An account of the complete history is given in [Grü95, Tou99b]. Among the main contributions are the following. In 1973, Joss and Shannon first proved that a quadrilateral can be constructed which requires k flips to convexify, for any chosen integer k. Thus the number of necessary flips is not a function of the number of polygonal edges. This proof was unpublished for a long time and appears in [Grü95]. Grunbaum and Zaks [GZ01] proved that a finite number of flips will also convexify a self-crossing polygon (note that the definition of a pocket must be adjusted to do this). Toussaint [Tou99b] showed that if we are also allowed to keep a pocket fixed and flip the remaining polygon, then polygons can be convexified in a finite number of flips using $O(n)$ time per flip. In 1973, Sallee [Sal73] proved that every non-convex continuous curve in $\mathbb{R}^d$ has a "chord-stretched" version with the same length, where the distance between any two points has not decreased. To do this, he first proved the result for polygons. In concluding remarks, he proposed the flip operation as a short intuitive alternate proof, stating that a countable number of flips suffice to convexify. He also mentioned that this idea was known to Choquet, from 1945 [Cho45]. The final stretched curve (or polygon) is planar and convex.

In 1993, Wegner [Weg93] considered the converse problem. He defined the *deflation* of a polygon, which is obtained by reverse-flips, and conjectured that any polygon can be completely *deflated* with a finite number of deflations. A polygon is deflated when it cannot be the result of a flip operation on some other polygon. Perhaps contrary to first intuition, Fevens et al. [FHM$^+$01] showed that there exist polygons with infinite sequences of deflations. They also showed that convexification is not always possible using *mouth-flips*. In this operation, flips are permitted only on adjacent edges meeting at a reflex vertex, even if the two edges do not define a pocket.

2.2.2 Flipturns

Along with their work on flips, Joss and Shannon invented a similar operation called the *flipturn* (see [Grü95]). Flipturns are sometimes called "inversions" in the physics literature. In this operation, a pocket is rotated by π and re-attached (see Figure 2.6).

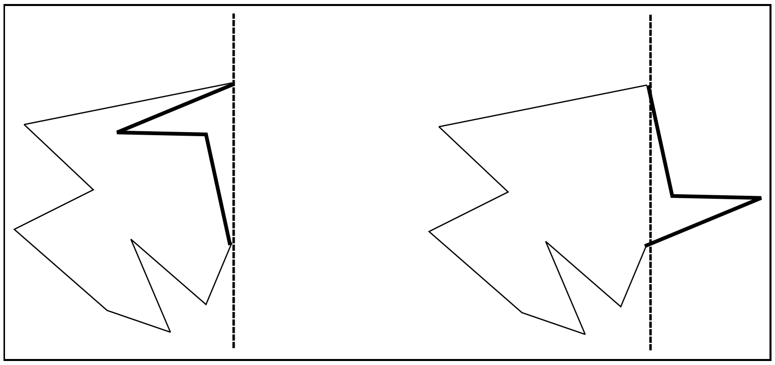

Figure 2.6: A flipturn: one pocket is rotated by π.

Joss and Shannon proved that any simple polygon can be convexified with at most $(n-1)!$ flipturns: after each flipturn the cyclic permutation of edges is changed and the polygon's area strictly increases. Thus no permutation is revisited. They also conjectured that $\frac{n^2}{4}$ flipturns suffice to convexify any polygon. Indeed, Ahn et al. [ABC$^+$00] came close by showing that any polygon will become convex after at most $n(n-3)/2$ flipturns. Note that there are two ways to interpret the "required number of flipturns for convexification". One may use carefully chosen flipturns, or instead arbitrarily (or poorly) chosen flipturns. Thus the result in [ABC$^+$00] applies to arbitrary choices. On the other hand, Biedl [Bie00] produced a polygon which requires at least $\frac{(n-2)^2}{4}$ poorly chosen flipturns to convexify. However, the same polygon can be convexified with $O(n)$ well chosen flipturns. The gap between linear and quadratic for well-chosen flipturns still remains. Note that the result of Ahn et al. actually includes the number of different edge slopes as a parameter. The bound given above is for the worst case of n distinct slopes. For orthogonal polygons the upper bound is actually $\frac{n}{2} - 2$ and is tight. To twist matters a bit more, Ahn et al. allow hull edges that are adjacent and collinear to pocket lids to be included in flipturns. As Aichholzer et al. [ACD$^+$02] pointed out, such degenerate conditions make all the difference, when we are concerned with the precise number of operations and not just asymptotics. They gave distinct results for different degeneracies and for "standard" conditions. All upper bounds for shortest and longest flipturn sequences for convexification of orthogonal polygons are linear and, to within a small constant factor, tight. The authors also gave quadratic bounds for all interpretations of flipturns, thus strengthening the results in [ABC$^+$00]. In addition, they provided the first non-trivial algorithms for computing convexifying sequences and gave NP-hardness results for finding shortest or longest sequences.

Finally, Grunbaum and Zaks [GZ01] proved that a finite number of flipturns will also convexify any self-crossing polygon.

2.2.3 Restrictions and extensions

In [ABD$^+$02], we considered a few extensions and similar operations to the flipturn. A *reversal* is similar to the flipturn, except that the rotated sub-chain need not be a pocket. A *transposition* exchanges the positions of two sub-chains. *Edge swaps* are reversals on sub-chains of size 2 (the analog to mouth-flips), and in *single-edge transpositions* one of the two exchanged sub-chains is a single edge and the other is empty (see Figures 2.7 and 2.8). Each of these operations is just a permutation of the edge list of a polygon or polygonal chain, and was inspired from sequence-permuting problems from the bioinformatics literature (see references in [ABD$^+$02]). Similarly, the definition of a *signed* polygon arose. In such a polygon each edge is given a direction (or parity), and this is affected by reversals. This parity affects the complexity results for integer permutations as well as polygonal permutations.

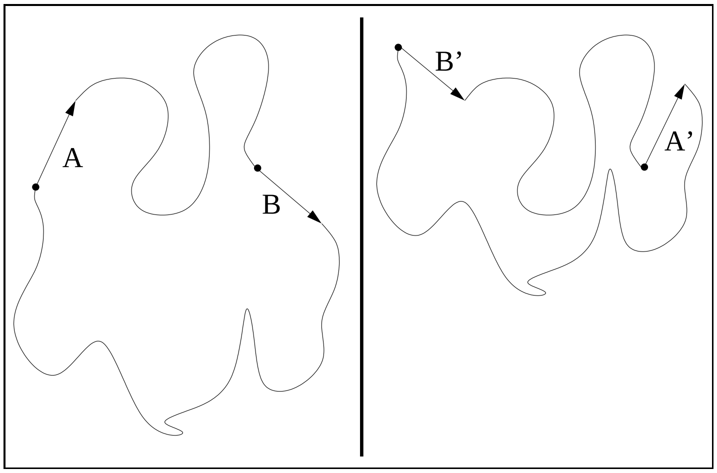

Figure 2.7: Transposing two edges A and B.

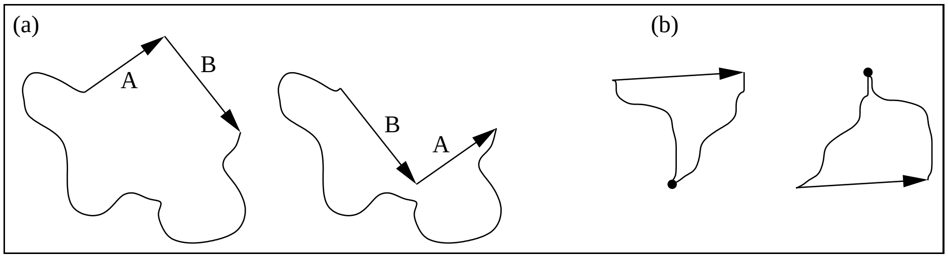

Figure 2.8: (a) An edge swap. (b) A single-edge transposition.

Little is known about permuting polygons, other than the results mentioned on flipturns. One interesting result is that if the longest edge of a polygon has unit length, then the polygon may be permuted to fit into a circle of radius $\sqrt{5}$ [GY79] (allowing crossings). In [ABD$^+$02] we showed that the two "local" operations, edge swaps and single-edge transpositions, are not sufficient to achieve any reconfiguration. Furthermore, determining whether a signed polygon can be permuted by transpositions so that its shape is rotated by π or so that it becomes a mirror image takes $\Theta(n \log n)$ time. NP-hardness results were given for using transpositions to fit a polygon inside a given rectangle, strip, or circle. Also, convexifying a star-shaped polygon with edge-swaps while maintaining star-shapedness can be done with $O(n^2)$ moves, and in the worst case this bound is tight. This last result was rediscovered by Ballinger [Bal03], who gave precise constants.

These studies of polygonal permutations were somewhat of an inspiration for Chan et al. [CGLQ03], who used simple properties of transpositions in an unexpected application: computing the optimal trajectory of a blind search, that starts at an unknown location in a river of known width, and that is guaranteed to find the river shore. So, as promised in Chapter 1, there is a connection between polygonal reconfigurations and foggy nights on rivers.

2.2.4 Motivation

The convexifying flip and flipturn operations have been applied mostly to generating random self-avoiding walks, though this is done mainly for lattice polygons with unit-length edges. Physicists, chemists and knot-theorists have been interested in random walks for some time now. These walks can be used to simulate polymer backbones, and help to predict various properties in statistical mechanics (see [DORS88]). Also, for the purposes of simulating physical knots, there is interest in enumerating knotted random walks with n steps (or equivalently, finding the probability that such a walk is knotted).

Standard methods for generating random self-avoiding walks on lattices iteratively choose a random new step and check if this causes an intersection. Such methods suffer from "attrition": if a random n-step walk crosses itself before n steps are completed, then it must be discarded, since backtracking creates a biased distribution. Even worse is the situation for closed walks; not only must there be no crossing, but the random walk must return to its starting point after exactly n steps. Otherwise it is useless.

To replace these methods, a brand new approach using *pivots* was invented. Most credit for this is given to Lal [Lal69] (see also [MS88]), who worked on the triangular lattice and allowed consecutive edges to turn only as if they were embedded on the hexagonal lattice. The pivot operation reflects the sub-chain between some edge e and an endpoint about the line through e. Thus, starting from a valid self-avoiding walk, one can randomly select an edge and apply a pivot, hoping to obtain a new walk. How often this succeeds in producing a new non-crossing chain, and how many iterations should be performed to minimize correlation are matters of intense study, but one thing is certain: the method works much better than previous techniques.

Lal's definition works well on the triangular/hexagonal lattice but there is a problem when it comes to the square, cubic, or hypercubic lattice. This is the lack of *ergodicity*, which means that not all random walks can be obtained by repeated uses of the pivot on some starting walk (and its successors). For example, a straight chain cannot be transformed at all. This certainly invalidates the technique as a true random generator. To fix this, Madras and Sokal [MS88] proposed to define the pivot on vertices instead of on edges. They also augmented the operation by allowing 90° rotations. This makes the pivot ergodic and the statisticians happy. Finally, they also generalized the pivot by including *diagonal reflections*, i.e. reversals.

Inspired by the work in [MS88], Dubins et al. [DORS88] worked on the self-avoiding closed walk, or simple orthogonal polygon. The lack of endpoints means that the pivot operations are now more restricted. Essentially, we can only use generalized flipturns (reversals), and only in some cases, flips. In other words, starting from a closed walk, the main method of generating a new one is to select two non-adjacent vertices and rotate one of the chains by π. Dubins et al. proved ergodicity by showing that any orthogonal polygon can be reconfigured to a rectangle with $2n-6$ flipturns (a bound improved by Ahn et al., as described earlier). It is a trivial matter to reconfigure between rectangles. Similar simple bounds were given for straightening n-step walks.

MacDonald et al. [MJHS85] had similar results to those of Madras and Sokal. They also described the "reptation" method of Wall and Mandel, where a chain is reconfigured by removing the last edge and placing it at the front. This is quite similar to the "single-edge transposition" defined in [ABD$^+$02].

Janse van Rensburg, Whittington and Madras [JWM90] proved that the pivot algorithm for polygons is ergodic on the face-centred cubic lattice. As one step, they showed that a polygon that has a convex projection can be flattened. These polygons with special projections and which are used as canonical configurations are revisited in Section 2.3.

Millett [Mil94] considered equilateral polygons in 3D for the purposes of geometric knot theory. Seeking a method to generate random knots, he rediscovered a variant of the flip operation. Two vertices are randomly selected, and one chain between them is rotated by a random angle about the line through the two vertices. To prove ergodicity, Millett demonstrated that a canonical configuration can be reached by first making a knot planar and star-shaped, and then convexifying it within the plane. The second step was done using mouth-flips, and it was shown that a finite number sufficed. Notice that since the polygons are equilateral, mouth-flips are identical to edge-swaps in this case. Thus, as Toussaint [Tou99b] pointed out, the arguments of Joss and Shannon apply, so $(n-1)!$ operations suffice to complete the second step. In fact, a better bound is $O(n^2)$ from the results of [ABD$^+$02]. Millet's technique was used in [DEJ97]. However, Toussaint also found that the first stage of Millett's ergodicity proof does not always terminate successfully, and proposed a simple remedy.

2.3 Universal motions of linkages

A linkage is a geometric graph with flexible joints. The universal model of motion is easy to visualize: as long as edge lengths and connectivity are preserved, every edge is allowed to move freely.

2.3.1 Early studies of general linkages

One of the earliest proofs known for linkages of n edges is that of Cauchy, from 1813 [Cau13]. Cauchy wanted to prove that if all the joints of a convex chain are opened (instantaneously) by some amount, then the distance between the endpoints increases (this is a generalization of Euclid's basic lemma for $n = 2$). Cauchy's lemma was part of a proof on the rigidity of convex polyhedra. Though the lemma does hold, the induction proof used had an error. This was noticed and corrected nearly a century later, by Steinitz (see [SR76]). Hence the lemma is found as a combination of these two names in the literature. The clearest and shortest proof is considered to be that of Schoenberg and Zaremba in [AZ99] (originally found in [SZ67]).

In Cauchy's "arm lemma", the smaller angle at each vertex is increased, but not beyond π. In other words if the chain makes only right turns, it still does in the final configuration. In fact, it is true that the distance between the endpoints of a chain increases as long as the minimum angle made at each joint is increased, regardless of what types of turns are made in the final configuration. This extension to Cauchy's lemma was formalized by O'Rourke [O'R01].

General planar linkages have been widely studied in the last couple of centuries, with immediate application to transferring mechanical motion. Edge-crossings were not an issue of concern, since many mechanical systems can be constructed on closely packed parallel planes to avoid any problems (think of the steam-powered train). Among the most well-known general linkages are those by Peaucellier, Watt, and Kempe. Peaucellier's linkage converts circular motion to straight line motion (beforehand, this had been finely approximated by Watt), and Kempe's constructions are theoretically possible to trace any polynomial curve. As is commonly said, "there exists a linkage that can sign your name". Further information and references are provided in [ST02, DOar]. For many small values of k, one can find a whole field of study for the k-bar (cyclic) linkage. For example, see [Tou03] for more details on the 4-bar linkage, its applications and uses within other geometry problems. The first algorithmic approach to linkage problems in the field of Computer Science appears to be that of Hopcroft, Joseph and Whitesides [HJW84], who studied the complexity of determining whether a designated joint can reach a given position. The work of Lenhart and Whitesides [LW95], or of Kapovich and Millson [KM95] is more directly related to the issues in this thesis. Independently they proved that all the configurations of an n-bar linkage are connected if, and only if, the second and third longest edges are each longer than the sum of the rest. Otherwise, there are exactly two connected components, which are symmetric (think of reflecting a triangle). See [KM95] for references to other geometers who were aware of this, though their publications are not widely available. An additional contribution in [LW95] is an algorithm to compute the reconfiguration between two polygons using $O(n)$ *line-tracking motions*. Each such motion moves one vertex along a straight line, while adjacent vertices act as elbows. The authors also discuss the literature in the field of motion planning, where linkages must reconfigure in the presence of obstacles. Whitesides [Whi01] surveys various linkage reconfiguration problems that have been considered in the past few decades, including polygonal reconfiguration within restricted spaces.

Amazingly, it is known that chains, trees and polygons in dimension 4 and above cannot lock [CO01]. It seems that we live in the most interesting dimension.

2.3.2 Non-crossing linkages in the plane

Aichholzer et al. [ADE$^+$01] showed how to reconfigure between two convex polygons using a linear number of simple moves without resorting to a canonical configuration. A simple move is one that affects only a constant number of vertices. The overall motion given is also "smooth" in the sense that each angle varies monotonically throughout. Biedl et al. [BDL$^+$99] gave an algorithm that runs in $O(n^2)$ time to produce a sequence of $O(n^2)$ simple moves that convexify any planar monotone polygon. Everett et al. [ELR$^+$98] gave an $O(n^2)$ time algorithm to compute $O(n)$ *complex* motions (i.e. each motion affects all joints) that convexify a star-shaped polygon.

Biedl et al. [BDD$^+$02] showed that some n-link trees can have configurations in $2^{\Omega(n)}$ distinct components of space. In an exploration of some rather advanced techniques, Connelly, Demaine and Rote [CDR02a] found that there exist locked trees with only one vertex of degree three.

One of the most important results of this entire section is that of Connelly, Demaine and Rote [CDR02b]: every simple polygon can be convexified and every simple chain can be straightened (this solves the puzzle in Figure 1.1). The result came after many efforts and a continuously shifting consensus on whether locked chains would be found or not (see their paper for more details). The key to the proof is the existence of a local infinitesimal motion (at any time), during which no vertices move closer together. Tools from many disciplines were used, including rigidity theory. The proof does not lend itself to analysis of algorithmic complexity. Following this result, Streinu [Str00] showed that in fact $O(n^3)$ motions suffice to convexify a simple chain[3] . Each motion has one degree of freedom, but takes an exponential amount of time to compute. The latest and most efficient algorithm for this problem computes a sequence of linear motions, the number of which is polynomially bounded by the

[3] A quadratic bound is given in the paper but it is an oversight, as pointed out by Streinu (personal communication).

size of the chain (n) and the ratio of maximum edge length to minimum separation (r) [CDIO04]. Specifically, the bound is $O(n^{79}r^{26})$. Luckily, each motion can be computed in quadratic time.

As should become apparent in the following section, planar polygons are useful as canonical configurations for three-dimensional linkages. In this context, the restriction to keep a polygon planar during reconfiguration is unnecessary. On the other hand, it makes sense if a planar configuration represents the projection of a three-dimensional polygon. Another use for planar reconfigurations is in graphics (see the discussion on polygon morphing at the end of this chapter).

2.3.3 Non-crossing linkages in $\mathbb{R}^3$

Biedl et al. [BDD+01] proved the existence of locked chains, with as few as five edges. They also proved that if an open chain has a simple orthogonal projection onto some plane, it can be straightened with $O(n)$ simple moves (note that the projection requirement can be determined in $O(n^4)$ time [BGRT99]). Their results were extended to chains embedded on convex surfaces. Cantarella and Johnston [CJ98] independently came up with the same locked chain. There is also a simple criterion to decide whether a chain with five edges can lock: it is only possible if the first and last edge lengths (separately) are longer than the sum of the middle three.

Alt et al. [AKRW04] proved that it is PSPACE-hard to decide whether two configurations of a chain are connected. Yet, the complexity of deciding whether a chain can be straightened is unknown. The interlocking of multiple short linkages was studied by Demaine et al. [DLOS03].

For polygons in $3D$ that happen to be planar, the flip operation can be clearly visualized as a method of convexifying which follows the rules of the universal model, as long as it is interpreted as a continuous rotation of a pocket about its lid. Recall that the number of moves is unbounded if the polygon is to remain planar after every flip. Aichholzer et al. [ADE$^+$01] worked on a pivot method to reconfigure between two planar convex polygons, generalizing the ideas of Millett [Mil94]. Each pivot affects only two angles, yet even in the case of convex polygons, an unbounded number of operations may be needed. Biedl et al. [BDD$^+$01] gave an algorithm using $O(n)$ simple moves. Aronov, Goodman and Pollack [AGP99] provided their own proof for this and remarked that their algorithm also works for crossing polygons.

The results above merely use the third dimension as a means of convexifying planar polygons. But most interest is in fully 3-dimensional polygons. We must immediately make an important observation for these objects: in R^3, polygons may be *knotted*. Thus reconfiguration questions only make sense within each knot class (i.e. within each connected component).

Randell [Ran94] examined the minimum number of edges necessary to form simple geometric knots. For example, he proved that the figure-eight knot requires at least seven edges. Only the trefoil knot may be drawn with six edges. With fewer than six edges, no knot can be drawn, other than the trivial knot (or *unknot*) which is continuously deformable to a loop. Randell recognizes independent discoveries of this lemma by Kuiper (unpublished) and by Millet [Mil94]. Millett (also recognizing collaboration with Orellana), showed that equilateral hexagonal unknots belong in one component. Principal references for these concepts are [Ran88b, Ran88a]. Millet's work was used in [DEJ97], as part of a simulated-annealing method of searching for the minimum of a knot-energy function. This minimum represents a "nice" configuration for a particular knot type.

A natural question to ask is whether all configurations of a polygon are connected, provided they are of the same knot type. By topological arguments alone, it is clear that this is not true for some knot types that have mirrored (*chiral*) configurations, such as the trefoil. The trefoil has left and right handed versions which are topologically different.

Calvo [Cal01] showed that even if edge lengths can vary, hexagonal trefoils are found in four components (two left-handed and two right-handed), as long as edges are given an orientation and one vertex is considered to be a special “root”. He also showed that equilateral hexagonal trefoils exist in four separate components.

In addition to their locked chain example mentioned earlier, Cantarella and Johnston [CJ98] showed that there exist two stuck hexagonal unknots for a set of specific edge lengths. The two examples are left and right hand versions of the same shape, and are illustrated in Chapter 5. Thus hexagonal unknots were shown to exist in three separate components. Cantarella and Johnston also proved that if the edge lengths were allowed to vary in a reconfiguration, then the separate components would merge. A stuck unknot (with 10 edges) was also found by Biedl et al. [BDD$^+$01] by doubling the edges of their locked chain. The property of fixed edge lengths, and in particular the ratio of lengths, is critical in these problems.

Toussaint [Tou01] found two more classes of unknots, beyond those of Cantarella and Johnston. They are illustrated in Chapter 5, where it is shown that there exist hexagonal unknots with nine separate components.

All these results extend to knots with an arbitrary number of edges, because in each example there is a short edge which can be broken into smaller ones without affecting the proofs. Cantarella and Johnston suspect that the number of stuck unknots expands geometrically as a function of n, but no bound is known. They also suspect that all equilateral unknots are in one component. This interesting problem has not yet been resolved.

Knowing that polygons in $\mathbb{R}^3$ can lock, all we can hope for is to classify as generally as possible the types of polygons that are not locked. Calvo et al. [CKM$^+$01] proved that if a polygon has a simple projection on some plane, then it can be convexified. In fact their algorithm to do this takes $O(n)$ time if the projection is convex. As they point out, the remaining time complexity of their algorithm depends on how fast a simple planar polygon can be convexified, since their proof depends on the result of Connelly, Demaine and Rote [CDR02b].

Mathematicians and physicists have been interested in knots as early as the 1860's. An informal account of this, as well as a discussion of the connections between knots and DNA is given in [FK97]. Sumners and Whittington [SW88] discuss the importance of knots in polymer chains, and use the self-avoiding walk method described earlier to study the probability of knottedness. Finally, according to Biedl et al. [BDD$^+$01], knot reconfigurations are also useful in the contexts of mechanical design, robotics, animation, rigidity theory and algebraic geometry.

2.4 Dihedral motions of linkages

If we extend the universal model by restricting angles to be fixed, we obtain the dihedral model of motion. The resemblance of this model to the "ball and stick" molecular model, used in introductory chemistry classes, is straightforward. A basic dihedral motion is defined on a selected edge. The entire linkage on one side of the selected edge is rotated rigidly about the axis of the edge (see Figure 2.9). This maintains all angles fixed. Notice that some of the pivot operations mentioned earlier, resembling flips on chains or orthogonal polygons, are specialized dihedral motions.

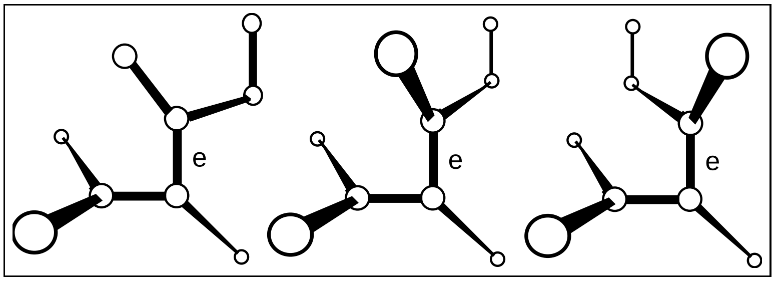

Figure 2.9: Three snapshots of a dihedral motion about edge e.

Analytic studies of the dihedral model were carried out in the chemistry community as early as the 1930's and 40's, by Eyring, Benoit and Taylor, each of them independently (see [Sos01, ST02] for details and a general survey). Since then the model has been used to generate random conformations of molecules and compute average properties of polymer chains (e.g. [SG72, FH76, FHK$^+$96]).

Algorithmically, there is not much to speak of prior to recent results by Michael Soss and colleagues. For example, Soss and Toussaint [ST01] developed a quadratic time algorithm to determine if the dihedral rotation about one edge results in edge-crossings. The lower bound given is $\Omega(n \log n)$, and this was nearly matched in the special case where the rotation is a full revolution. Soss, Erickson and Overmars [EOS03] showed that pre-processing a dihedral chain hardly helps for computing such rotation queries. Specifically, either the preprocessing time or n such queries *almost certainly* require $\Omega(n^2)$ time[4]. As for general reconfigurations, Soss and Toussaint [ST01] showed that deciding whether a chain can be flattened is NP-hard [Sos01]. All of the above appear in the recent doctoral thesis by Soss [Sos01]. He proved additional NP-hardness results for computing maximum endpoint distance and determining whether chiral (mirror image) reconfigurations are possible. Finally, multiple interlocked short chains were illustrated.

Recently, Demaine, Langerman and O'Rourke [DLO03] considered non-acute chains that can be "produced" through the apex of a cone. Minor details left aside, a chain is defined to be producible up to an edge e, if it can avoid self-intersection outside of the cone, while e is still inside and being pushed out through the apex. Edges beyond e are not considered, as they are not yet produced. The cone angle limits the allowable range of each fixed dihedral angle in a straightforward way. For chains satisfying the limiting range, it is shown that they are producible if and only if they can be flattened.

A main open problem left from [Sos01] is that of determining whether all flat

[4] This bound can be broken only if the same happens for a large class of problems for which no sub-quadratic time algorithm is known.

configurations of dihedral chains are connected. Many problems regarding *flat-state connectivity* were solved in [ADD$^+$02, ADM$^+$02], by imposing additional restrictions to polygons, chains and trees. These are stated in Chapter 6. Also in that chapter is a detailed proof that chains with unit-length edges, and with angles within the range $(60°, 150°)$ are flat-state connected.

2.5 Folding and unfolding polyhedral surfaces

A polyhedron is an object with a boundary consisting of flat polygonal surfaces, also known as faces. Every edge on the boundary of a face also belongs to the boundary of another face. Faces do not intersect other than at common edges or vertices. Apart from endpoints, every edge belongs to exactly two faces. Finally, there is a path along the surface between any two points.

Note the subtle point, that each polygonal face is regarded to be a *rigid* surface (at least in this thesis). Assume that the interior of each face is constructed of a material that cannot be deformed. If this were not the case, we would be dealing with a 3-dimensional linkage. For a great introduction to polyhedra, including a history of their appearance and use since ancient times, see the book by Cromwell [Cro97].

The description given above for polyhedra resembles that of simple polygons, if we replace “faces” with “edges”, and “edges” with “vertices”. However things become a little more complicated in the case of polyhedra, since they can topologically resemble a sphere, doughnut (torus), or other classes of surfaces. Unless mentioned otherwise, it will be assumed that a polyhedron resembles a sphere and is convex. The definition of convexity is directly analogous to that for polygons.

In 1813, Augustin Cauchy proved that polyhedra are rigid objects [Cau13, AZ99] [5]. Cauchy’s theorem in fact states that if two convex polyhedra are combinatorially equivalent and all corresponding faces are congruent, then the two polyhedra are

[5] Recall that Cauchy’s “arm lemma” was incorrectly proven as part of this rigidity theorem.

congruent. Thus, once we specify the combinatorial arrangement and exact shape of each face, there is only one configuration that the polyhedron can have. This theorem does not hold if non-convexity is allowed. One can construct a cube, replacing the top face with a pyramid which could either form a "roof" or a "dent", thus creating two different (but combinatorially equivalent) polyhedra. For a long time it remained open whether there exist flexible non-convex polyhedra. There was evidence to support the conjecture that none exist. However, Connelly [Con78] discovered a counterexample. Cromwell [Cro97] summarizes the history that led to this discovery. An amusing detail is quoted here:

"At a 1975 topology conference in Cornell University, Connelly heard that another mathematician, working in a different area of mathematics, had been working on the same problem and had found a flexible polyhedron. Disheartened by the news, but curious to find out more, Connelly traced the rumour back to its source. Eventually he discovered that it referred to himself!"

Given the rigidity of convex polyhedra and the relatively rigid nature of non-convex polyhedra, it may seem that few reconfiguration questions can be asked here. On the contrary, a classic open problem is whether every convex polyhedron can be cut along some of its edges and "unfolded" to a planar polygon without overlap. Apparently, this problem dates back to the 16th century, being implicit in the work of Albrecht Dürer [Dür77] (see [Cro97]). The problem was stated again in modern times in [She75] (see also [O'R00]). An example of an edge-unfolded cube is shown in Figure 2.10.

According to O'Rourke [O'R00], almost all random unfoldings of random polyhedra overlap, yet it is difficult to construct such examples manually. It is still widely conjectured that for every convex polyhedron there exists an appropriate set of edges to cut. For non-convex polyhedra, it is easy to construct examples that cannot be edge-unfolded (take a large cube with a smaller cube sticking out of one face). In fact, even if we require all faces to be convex (or even triangular), Bern et al. [BDE$^+$03]

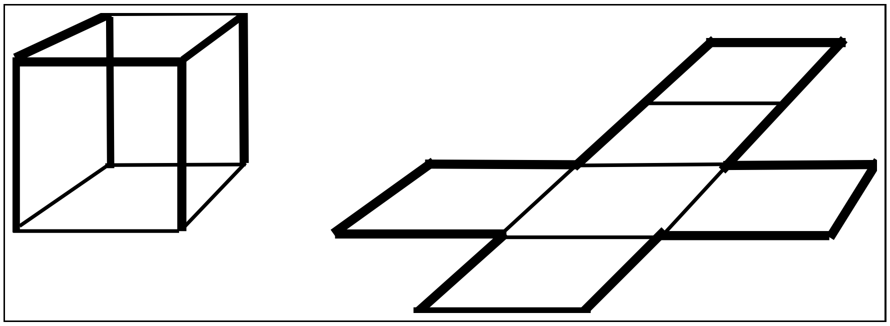

Figure 2.10: Edge-unfolding a cube. Bold edges are cut.

constructed examples of “ununfoldable” non-convex polyhedra.

On the other hand, if cuts are allowed to be made through faces, it is known that every convex polyhedron can be unfolded. For example, *star unfolding* has been used since the late 80’s in methods for finding geodesic paths on polyhedra (see [AAOS97]). A star unfolding is obtained by finding a point that has n unique shortest paths to the polyhedron vertices, and cutting these paths. Clearly, since every vertex is affected by a cut, this allows the polyhedron to unfold. Aronov and O’Rourke [AO92] proved that star unfoldings do not overlap. The *source unfolding*, also known to not overlap, involves the set of points with more than one shortest path to a common source (see [AAOS97, DOar] for more details). Biedl et al. [BDD$^+$98] showed that some classes of orthogonal non-convex polyhedra may be unfolded by cutting across faces.

More closely related to edge unfolding is the *vertex unfolding* problem. This is just a weaker formulation, where cuts are still made on polyhedral edges but the regions of the resulting planar polygon are allowed to be connected through vertices. For example, it is allowed to cut all edges of a polyhedral face, as long as one vertex remains attached to a neighbouring face. It is known that all triangulated polyhedra, convex or non-convex, can be vertex-unfolded [DEE$^+$02]. For non-triangular faces, the problem is still open.

The opposite problem to those discussed so far is that of folding a polygon into a polyhedron. Suppose that each edge must match up and be glued to only one other edge. Even if interior creases are not pre-specified (i.e. diagonals that will end up becoming polyhedral edges) there is a quadratic time and space algorithm that decides if a given polygon can fold, by Lubiw and O'Rourke [LO96]. They used an existence theorem by Aleksandrov, who gave simple sufficient conditions to fold a polygon (ignoring the edge matching requirement). The number of possible foldings and unfoldings between polygons and polyhedra is examined in [DDLO02].

Returning to the original classic problem of edge-unfolding polyhedra, consider this related definition and problem posed in the late 90's[6] . Define a *band* to be the surface of a convex polyhedron, enclosed between two parallel planes and not containing vertices of the polyhedron. Can every band be unfolded by cutting exactly one edge? A related and easier question is whether any band can be unfolded by cutting the shortest path between the two planes. This was conjectured to be true by O'Rourke in 2000 (see [O'R03]). In Chapter 7 the band unfolding problem is solved, and arguments are given for the proof of O'Rourke's conjecture, based on the proof for bands.

With a little imagination, one can see that cutting an edge of a band yields a polyhedral surface which contains a series of "hinges" that can rotate. Such constructs have been studied before. Bhattacharya and Rosenfeld [BR95] define a polygonal *ribbon* as a finite sequence of polygons, not necessarily coplanar, such that each pair of successive polygons intersects exactly in a common side. Triangular and rectangular ribbons (both open and closed) have also been studied. Simple bands can be used as linkages to transfer mechanical motion, as pointed out by Cundy and Rollett [CR61]. Singularities of such "panel and hinge" structures were examined by Borcea and Streinu [BS05]. In fact, in their analysis, bands are three-dimensional generalizations of planar universal-motion chains. Last, but not least, continuous

[6] Though not stated in published form, this question has been circulating since at least 1998, apparently first posed by E.Demaine, M.Demaine, A.Lubiw and J.O'Rourke.

ribbons were studied by Sir Francis Crick as representations of DNA strands [Cri76]. In all these instances, almost no attention was paid to questions regarding the non-self-intersecting states or the collision-avoiding motions of bands.

Finally, applications of folding polyhedra and linkages, mentioned throughout the literature, are in bending sheet metal or hydraulic tubes. In the manufacturing business, there is interest in constructing $3D$ polyhedral surfaces (not necessarily closed polyhedra), by creating creases on planar polygonal sheets of metal and then folding (see [AFM03, O'R00] and references therein). Naturally, if one has a $3D$ shape in mind, a good question is whether it can actually be constructed from one piece of sheet metal. This is equivalent to asking whether it can be unfolded without overlap. Realistically, these applications are likely to be impeded primarily by motion planning problems between machinery and product, and are likely to involve non-convex polyhedra with relatively simple shapes and few edges (up to a few dozen). Still, as illustrated in Chapter 7, very simple shapes can easily lead to trouble if we are not careful.

2.6 Other reconfigurations

H-P model: An interesting lattice model of protein folding is the H-P model, where H and P stand for *hydrophobic* and *hydrophilic* amino-acids respectively. In this model, a typical task is to find the minimum energy configuration of an H-P chain, where vertices are labelled H or P and edges have unit length. Mimicking the tendency of amino-acids to either avoid or seek water, minimum energy is defined so as to maximize the number of H-adjacencies on the lattice. The paper by Aichholzer et al. [ABD$^+$03] provides a detailed introduction and the most recent results for this model.

Origami: Paper folding (origami) is somewhat similar to polyhedral unfolding. It involves reconfiguring and folding a surface that cannot tear, stretch, or cross itself (just think of very durable paper). On the other hand, this model allows self-intersection and in fact sometimes demands it. Also, paper is allowed to bend, and creases may be formed anywhere. The typical issue of interest is to determine whether a given shape can be constructed from a flat polygonal piece of paper. A large variety of problems, including that of map folding, is discussed in [DOar].

Hinged sets of polygons: Consider two simple planar polygons that are attached at a single common vertex, which acts as a hinge. An allowed state is one where the polygons do not overlap. A "hinged set" (of polygons) is just a connected collection of such polygons and hinges. Clearly, such a hinged set may reach several configurations, via rotations at each hinge. Deciding whether two valid configurations are connected through hinge motions is PSPACE-hard [HDF03]. Extensions to sets of polyhedra hinged at edges have only recently been looked into [DDLS04].

Polygon morphing: Grenander, Chow and Keenan [GCK91] investigated morphing between polygons that have parallel edges (or equivalently, identical sequences of interior angles). They proved that by shifting edges, maintaining their directions and possibly extending others, reconfiguration is possible. In the same model, Guibas, Hershberger and Suri [GHS00] consider a morphing "step" to be a uniform scaling or translation of part of a polygon. They show that $O(n \log n)$ such steps are sufficient to morph, using $O(n \log n)$ time to compute the steps. It is interesting that the authors establish a form of correspondence between their transformations and rotations in binary trees, as is the case with edge flips in triangulations.

One way of viewing polygonal shape is to consider a tree as the "skeleton" of a polygon. For example, one can use the spanning tree of the dual of a triangulated polygon[7]. Then, polygonal morphing can become an issue of reconfiguring spanning trees. With this in mind, Aichholzer, Aurenhammer and Hurtado [AAH02b] proved

[7] The dual of any triangulation has one point representing each triangle and one edge between points that represent adjacent triangles. The spanning tree of a point set is a tree that covers all points.

that any non-crossing geometric spanning tree of a point set can be reconfigured to the minimum (length) spanning tree with $O(n \log n)$ length-improving *edge moves*. An edge move deletes one edge and inserts another and is useful for the enumeration of spanning trees. It is known that a linear number of edge moves (neglecting length improvement) suffice to reconfigure between spanning trees in general position [AF96]. The general position requirement was removed in [ABG$^+$04]. The *edge slide* is a more local operation, where one endpoint of an edge remains fixed and the other slides along a neighbouring edge. This operation, introduced in the combinatorial setting by Goddard and Swart [GS96], was investigated in [AAH02b] with the motivation of polygonal morphing and enumeration. Aichholzer and Reinhardt [AR05] proved that $O(n^2)$ edge slides suffice to reconfigure between any two spanning trees.

Hernando, Houle and Hurtado [HHH02] defined local transformations for the boundaries of polygons embedded on fixed point sets. Using their types of "flips", they show that all weakly externally visible polygons are connected[8] . They remark that no other local transformation is known to connect a more general class of polygons on fixed point sets.

Pach and Tardos [PT01] considered the problem of "untangling" a crossing polygon into a simple one. The operation used was a point move, i.e. the relocation of a point's coordinates. At most $n - \Omega(\sqrt{n})$ operations always suffice.

[8] A polygon is weakly externally visible if for every boundary point there is a ray to infinity that does not cross the remaining boundary.

Chapter 3

Similarity of Monotone Chains

In this chapter we focus on one of the simplest types of reconfigurations: rigid motions. The structures considered here are monotone chains. The question studied here is not whether one chain can be transformed to another, since that can be answered quite easily for rigid motions. Instead we are concerned with how *similar* two chains are, if we are allowed to move them rigidly. As described below, the field of *melodic similarity* provides us with a measure of chain similarity and some interesting motivation for our problem. Most previous work in the area of motivation has involved approaches not related to polygonal reconfigurations. Thus no background was presented in Chapter 2 and we give a brief introduction in the following section.

3.1 Background and motivation

The manner in which humans detect similarities in melodies has been studied extensively [Mar01, HE02, MF04]. There has also been some effort in modeling melodies so that similarities can be detected algorithmically. Some results in this fascinating study of musical perception and computation can be found in a collection edited by Hewlett and Selfridge-Field [HSF98].

Similarity measures for melodies find application in content-based retrieval methods for large music databases such as *query by humming* (QBH) [GLCS95, MHK99] but also in other diverse applications such as helping prove music copyright infringement [Cro98]. Previous work on rhythmic and melodic similarity is based on methods like one-dimensional edit distance computations [Tou04], approximate string-matching algorithms [BNMW+99, Lem00], hierarchical correlation functions [LYZ01], two-dimensional augmented suffix trees [CCC+00], transportation distances [TGV+03, LT04], and maximum segment overlap [ULM03].

Ó Maidín [Ó98] proposed a geometric measure of the difference between two melodies, M_a and M_b. The melodies are modeled as monotonic pitch-duration rectilinear functions of time as depicted in Figure 3.1. This rectilinear representation of a melody is equivalent to the triplet melody representation in [LYZ01]. Ó Maidín measures the difference between the two melodies by the minimum area between the two polygonal chains, allowing vertical translations. The area between two polygonal chains is found by integrating the absolute value of the vertical L_1 distance between M_a and M_b over the domain θ. Arkin et al. [ACH+91] show that the minimum integral of any distance L_p ($p \geq 1$) between two orthogonal cyclic chains, (allowing translations along θ and z) is a metric.

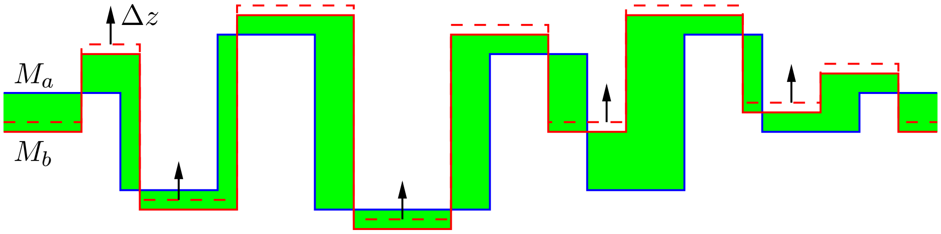

Figure 3.1: The area between two melodies, M_a and M_b, and a small vertical shift of M_b by Δz that changes the area.

In a more general setting such as music retrieval systems, we may consider matching a short query melody against a larger stored melody. Furthermore, the query may be presented in a different *key* (transposed in the vertical direction) and in a different *tempo* (scaled linearly in the horizontal direction). Francu and Nevill-Manning [FNM00] compute the minimum area between two such chains, taken over all possible transpositions. They do this for a constant number of pitch values and scaling factors, and each chain is divided into m and n equal time-steps. They claim (without describing in detail) that their algorithm takes $O(nm)$ time, where n and m are the number of unit time-steps in each query. This time bound can be achieved with a brute-force approach.

In some music domains such as Indian classical music, Balinese gamelan music and African music, the melodies are cyclic, i.e. they repeat over and over. In Indian music these cyclic melodies are called *talas* [Mor98]. Two such monophonic melodies may be represented by orthogonal polygonal chains on the surface of a cylinder, as shown in Figure 3.2. This is similar to Thomas Edison's cylinder phonographs, where music is represented by indentations around the body of a tin foil cylinder. We consider the problem of computing the minimum area between two such chains, over all translations on the surface of the cylinder.

We present two algorithms to find the minimum area between two given orthogonal melodies, M_a and M_b of size n and m respectively ($n > m$). The algorithms may be used for cyclic melodies or in the context of retrieving short patterns from a database (open planar orthogonal chains). We have chosen to describe the algorithms for the case where the melodies are cyclic. The first algorithm, given in section 3.2, will assume that the θ direction is fixed. The second algorithm, described in section 3.3, will find the minimum area when both the z and θ relative positions may be varied. In each case, we will assume that the vertices defining M_a and M_b are given in the order in which they appear in the melodies. In section 3.4 we discuss natural extensions, both for the polygonal description of melodies and for the types of queries.

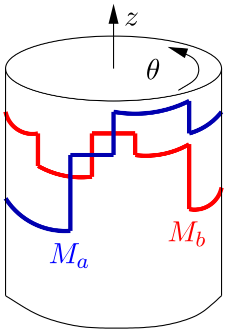

Figure 3.2: Two orthogonal periodic melodies.

3.2 Minimization with respect to z direction

In the first algorithm, we will assume that both melodies are fixed in the θ direction. Without loss of generality, we will assume that melody M_a is fixed in both directions, so all motions are relative to M_a. In Figure 3.1 we show the area between two melodies, and a small shift of M_b in the z direction.

To see how the area between the two melodies changes as M_b moves in the z direction, consider a set of lines defined by all vertical segments of the melodies as shown in Figure 3.3. This set of lines partitions the area between the melodies into rectangles C_i, $i = 1, \ldots, k$, each defined by two vertical lines and two horizontal segments, one from each melody. Note that k is at most $\frac{n+m}{2}$. The area between M_a and M_b is the sum of the areas of all C_i. If M_b starts completely below M_a and moves in the positive z direction, then for any given C_i the lower horizontal segment (from M_b) will approach the upper fixed horizontal segment while the area of C_i decreases linearly. This happens until the horizontal segments are coincident (and the area of C_i is zero). Then the upper horizontal segment (now from M_b) will move away from the lower fixed horizontal segment while the area of C_i increases linearly.

We will consider the vertical position of M_b to be the z-coordinate of its first edge. We define $z = 0$ to be the position where this edge overlaps the first edge of M_a. Let $A_i(z)$ denote the area of C_i as a function of z. Define z_i to be the coordinate at which

$A_i = 0$. These k positions of M_b where some A_i becomes zero will be called z-events. The slope of $A_i(z)$ is determined by the length of the horizontal segments of C_i. The total area between M_a and M_b is given by $A(z) = \sum_{i=1}^{k} A_i(z)$. Note that since $A(z)$ is the sum of piecewise-linear convex functions, it too is piecewise-linear and convex. Furthermore its minimum must occur at a z-event.

Theorem 3.1. *A minimum for $A(z)$ can be computed in $O(n+m)$ time.*

Proof. The function $A(z)$ is given by $A(z) = \sum w_i|z_{bi} - z_{ai}|$, where z_{bi} is the vertical coordinate of M_b in C_i, z_{ai} corresponds to M_a, and w_i is the weight (width) of C_i, as shown in Figure 3.3. Let α_i denote the vertical offset of each horizontal segment in M_b from z_{b1}. Thus we have $z_{bi} = z_{b1} + \alpha_i$, and $A(z) = \sum w_i|z_{b1} - (z_{ai} - \alpha_i)|$. Finally, notice that the term $z_{ai} - \alpha_i$ is equal to z_i. Thus we have $A(z) = \sum w_i|z_i - z_{b1}|$. This is a weighted sum of distances from z_{b1} to all the z-events. The minimum is the weighted univariate median of all z_i and can be found in $O(k)$ time [Rei78]. This median is the vertical coordinate that z_{b1} must have so that $A(z)$ is minimized. Once this is done, it is straightforward to compute the sum of areas in $O(k)$ time. Recall that k is at most $\frac{n+m}{2}$.

□

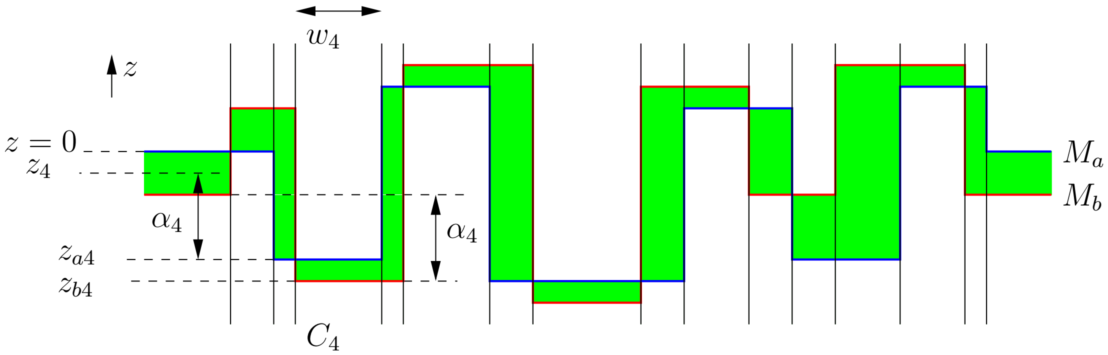

Figure 3.3: Contribution of C_4 to area calculation.

3.3 Minimization with respect to z and θ directions

If no vertical segments among M_a and M_b share the same θ coordinate, then M_b may be shifted in at least one of the two directions $\pm\theta$ so that the sum of areas does not increase. This means that in order to find the global minimum, the only θ coordinates that need to be considered are those where two vertical segments coincide. Thus our first algorithm may be applied $O(nm)$ times to find the global minimum in a total of $O(nm(n+m))$ time. We now propose a different approach to improve this time complexity.

As described in the previous section, for a given θ, the area minimization resembles the computation of a weighted univariate median. When we shift M_b by $\Delta\theta$, we are essentially changing the input weights to this median. Some C_i grow in width, some become narrower, and some stay the same width. As we keep shifting, at θ coordinates where vertical segments coincide, we have the destruction of a C_i and creation of another C_i. An important observation is that all C_i grow (or shrink) at the same rate.

Let us store the z-events and their weights in the leaves of a balanced binary search tree. Each leaf represents one C_i. The leaves are ordered by the value z_i. Each leaf also has a label to distinguish between C_i that are growing, shrinking, or unaffected when M_b is shifted infinitesimally in the positive θ direction. At every node with subtree T we store:

- W_T : The sum of weights of all leaves in T.
- D : The number of growing leaves minus the number of shrinking leaves in T.

The weighted median of all z_i may be calculated by traversing the tree from root to leaf, always choosing the path that balances the total weight on both sides of the path. The time for this is $O(\log k)$.

Suppose that we shift M_b by some offset $\Delta\theta$, which is small enough such that no vertical segments overlap during the shift. Each w_i belonging to a growing leaf must be increased by $\Delta\theta$, and each w_i belonging to a shrinking leaf must be decreased by this amount. Instead of actually updating all our inputs, we just maintain a global variable $\Delta\theta$, representing the total offset in the θ direction. The total weight of a subtree T is now $W_T + D\Delta\theta$.

When we shift to a position where two vertical segments share the same θ coordinate, we potentially eliminate some C_i, create a new C_i, or change type of C_i. The number of such changes is constant for each pair of collinear vertical segments. The weight given to a created leaf must equal $-\Delta\theta$. Each of these changes involves $O(\log k)$ work to update the information stored in the ancestors of a newly inserted/deleted/altered leaf. There are $O(nm)$ such instances where this must be done and where the median must be recomputed, so the total time to compute all candidate positions of M_b is $O(nm \log(n + m))$.

At every θ coordinate where we recalculate the median, we also need to calculate the integral of area between the two melodies. For a given median z_*, the area summation for those C_i for which $z_* > z_i$ has the form $\sum w_i(z_* - z_i)$.

This may be calculated in $O(\log k)$ time if we know the value of this summation for every subtree. In order to do this, we store some additional information at every subtree T. Specifically, the area is given by

$z_*(W_T + D\Delta\theta) - \sum(w_i z_i) - \Delta\theta \sum(I z_i)$,

where in the second summation I takes the values $(+1, 0, -1)$ for growing, unchanged and shrinking leaves respectively. These two summations are the additional parameters that need to be stored, and they may be updated in $O(\log k)$ time at every critical θ coordinate.

We must also perform a similar $O(\log k)$ time calculation of $\sum w_i(z_i - z_*)$, for all

$z_i > z_*$. No additional parameters are needed for this.

Since at every critical θ position we can calculate the median and integral of area in $O(\log k) = O(\log(n+m))$ time, we obtain the following theorem:

Theorem 3.2. *Given two orthogonal periodic melodies with n and m vertices, a relative placement such that the area between the melodies is minimized can be computed in $O(nm \log(n+m))$ time.*

The analysis above may be used for the problem of matching two planar orthogonal monotonic open chains. Clearly if we are only interested in varying one direction, an optimal placement may be found in linear time. If the direction of monotonicity is the x-axis, then this problem is more interesting if one of the two chains has a shorter projection onto the x-axis. This "shorter" chain reminds us of a short motif that we might search for in a larger database of music. For this problem, we measure area only within the common domain of the two chains along the x-axis. Naturally, the projection of the shorter chain must be entirely covered by the projection of the longer chain.

Corollary 3.3. *Given two planar orthogonal chains monotone with respect to the x-axis, with n and m vertices respectively, a relative placement such that the area between the chains is minimized can be computed in $O(nm \log(n+m))$ time.*

Arkin et al. [ACH$^+$91] showed that two polygonal shapes may be compared by parametrizing their boundary lengths and examining their orientation differences. They showed that their measure, which is invariant to scaling, rotation and translation, can be computed by finding the minimum integral of the vertical distance between two orthogonal chains, which are constructed in a preprocessing step (Figure 3.4). Specifically, every edge of a polygon corresponds to a horizontal segment of proportional length in a monotone chain. Every turning angle of a polygon corresponds to a vertical segment. In fact some of the techniques in [ACH$^+$91] are similar

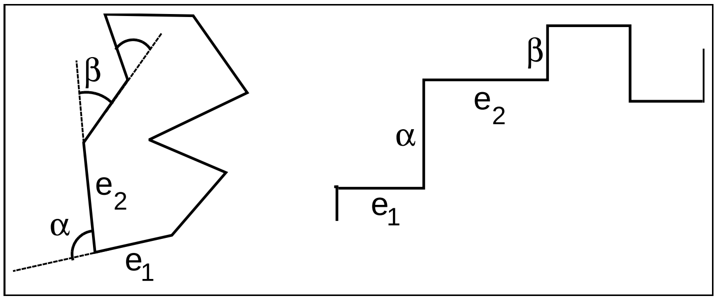

Figure 3.4: A polygon represented by an orthogonal chain.

to those given in this section. However, they chose to use the L_2 distance, as opposed to the L_1 distance used here. Using the L_2 distance, the optimal z-position at any θ can be computed in $O(1)$ time, so there is no need to use a data structure as was done in this chapter. The complexity of the algorithm given by Arkin et al. is dominated by sorting the $O(nm)$ critical θ events. They indicated that their algorithm offers no improvement over a $O(n^3)$ time brute-force approach for the L_1 metric.

3.4 Extensions

3.4.1 Higher dimensions

Consider a simple orthogonal open chain which is monotone with respect to the x-axis. Furthermore, at any particular x-coordinate suppose that the chain has at most two edges (in the y- and/or z-directions). This is an extension of the melody representation which we have seen so far. The x-axis still represents time, but perhaps now the other axes might represent pitch, loudness, timbre or chord density. In the plane, the measurement made was an integral of the pitch (height) difference taken over a domain in the x-axis. Here, we still wish to minimize an integral of the distance between two chains over all common x-coordinates. Whether this should be Euclidean distance or perhaps the L_1 distance is debatable. The latter is definitely easier to compute. Suppose that we only allow motions of the chains M_a and M_b in the y- and z-directions. Minimizing the sum of pairwise Euclidean distances is equivalent to the Weber problem, which involves finding a point with minimum sum of distances to points in a given set. It is not possible to find an exact solution to the Weber problem (also known as the generalized Fermat-Torricelli problem; see [GS98]). Using the L_1 metric, the function to minimize is $\sum w_i(|z_{bi} - z_{ai}| + |y_{bi} - y_{ai}|)$. This may be split into two terms, $\sum w_i|z_{bi} - z_{ai}| + \sum w_i|y_{bi} - y_{ai}|$. Thus we just have to make two univariate median computations to find the optimal (y, z) placement for a particular relative position of the two chains in the x-direction. In $\mathbb{R}^d$ we can accomplish this task in $O(dn)$ time. The decoupling of the two coordinates allows us to update each median separately at every critical x coordinate. In $\mathbb{R}^3$ there are still $O(nm)$ critical x coordinates and $O(n+m)$ weights/leaves, so the time complexity is the same as for planar chains. If we let n and m be the total number of edges parallel to the x-axis for two chains, then in $\mathbb{R}^d$ the time complexity becomes $O(mnd \log(m+n))$, using $O(dn)$ space. Note that only these edges are significant in any of the computations we have made so far.

3.4.2 Scaling

Here we consider the effect of scaling planar chains, either in the vertical or horizontal directions.

If we shrink the shorter chain horizontally, the domain of the integral becomes smaller, so the total area will tend to zero eventually. How should we deal with this? It seems reasonable to normalize by computing the total area over the domain of the smaller chain. It is equivalent to fix the shorter chain at unit domain length and modify the larger chain instead. Its domain would expand from unit length to some value where its narrowest strip has unit width.

Let an *x-value* be an x-coordinate where there are vertical segments from both chains.

Lemma 3.4. *For the scaling method proposed above, the optimal scaling of the larger chain occurs at a position where two or more x-values occur.*

Proof. For a particular scaling value we know that the optimal placement of the larger chain occurs when we have an x-value. This follows from the arguments given in section 2. Suppose that somehow we know the optimal scaling factor. Assume that there is only one x-value and we know which two vertical segments are aligned. Now we can keep scaling the large chain while using the x-value as an "anchor". One of the two scaling directions will improve the area minimization, at least until we obtain another x-value. □

This means that we have $O(n^2m^2)$ candidate configurations, so a brute-force algorithm would take $O(n^3m^2)$ time using $O(n)$ space. The lemma also applies to vertical scaling. In this case a brute-force algorithm would have a time complexity of $O(n^3m^3\log(n+m))$.

3.4.3 Non-orthogonal chains

In the preceding sections it is assumed that a melody may be divided into intervals, and within each interval the pitch (or volume/timbre) remains constant. In a more general setting, these features may vary within each interval. A further step in this direction is to consider monotonic piecewise linear chains. Consider two such planar chains. Let us divide the plane into strips, just as we had for orthogonal chains. In this case, a vertical boundary is placed at every vertex, as shown in Figure 3.5.

Thus within every strip we have two linear segments. Suppose we vary only the relative pitch of the chains. As one chain is moved down from infinity, within a given strip the area decreases linearly until the two segments touch inside the strip. Then the area decreases quadratically until the midpoints of the segments intersect.

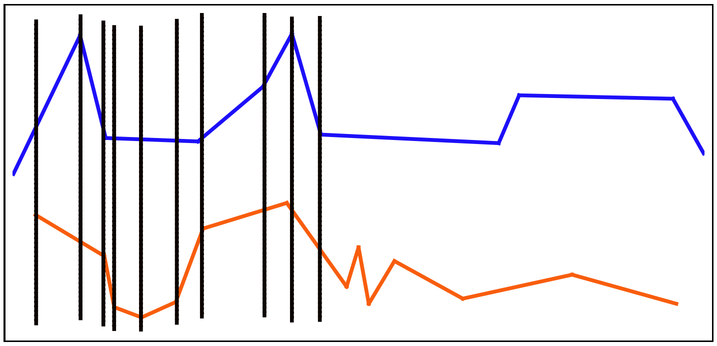

Figure 3.5: Two monotone chains and some of their strips.

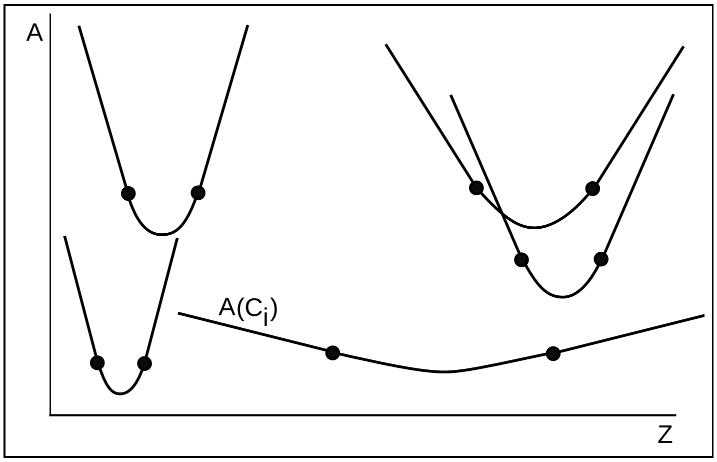

Figure 3.6: A set of area functions from the C_i strips.

Of course, the reverse occurs as we keep moving the chain down. The overall area function of each strip C_i is now a symmetric convex function, which is part linear and part quadratic (around the symmetric point). The total area is a sum of n functions, such as those shown in Figure 3.6.

The area function is convex and piecewise quadratic with $O(n)$ inflection points. Specifically, in the aggregate function an inflection point will exist only at a coordinate where some individual function changes from linear to quadratic. There are two such points per individual function. Note that the minimum of the aggregate function need not occur at an inflection point, unlike the case of orthogonal chains. Now, it is possible for the minimum to exist between two consecutive inflection points. This would be the only region between two successive inflection points where the function is not monotone.

To compute the minimum of the aggregate function, we give the following algorithm:

1. Let R be the set of individual area functions. Let F be a single quadratic term, initialized at zero.

2. Compute Q_1, the median of the x-coordinates of the minima of all functions in R, as shown in Figure 3.7.

3. Compute the value and gradient of the total area function at Q_1, by querying F and all functions in R. If not at the global minimum, assume without loss of generality that the minimum is to the left of Q_1.

4. For the subset of functions in R whose minima are to the right of Q_1, compute the median Q_2 of their *left* inflection points. Q_2 splits the subset into the *left* group and the *right* group.

5. If $Q_2 \geq Q_1$, as shown in Figure 3.8, replace all functions in the *right* group with a single linear term, which is a summation of all individual left-hand linear terms. Update F by adding this term to it. Remove the *right* group from R.

6. Else if $Q_2 < Q_1$, as shown in Figure 3.9, compute the gradient of the total function at Q_2. If the global minimum is to the left of Q_2, follow the instructions of step 5 on the *right* group. Otherwise if the minimum is between Q_2 and Q_1, replace all functions in the *left* group with a single quadratic term, which is a summation of all individual quadratic terms. Then update F and remove the *left* group from R.

7. Go to step 2.

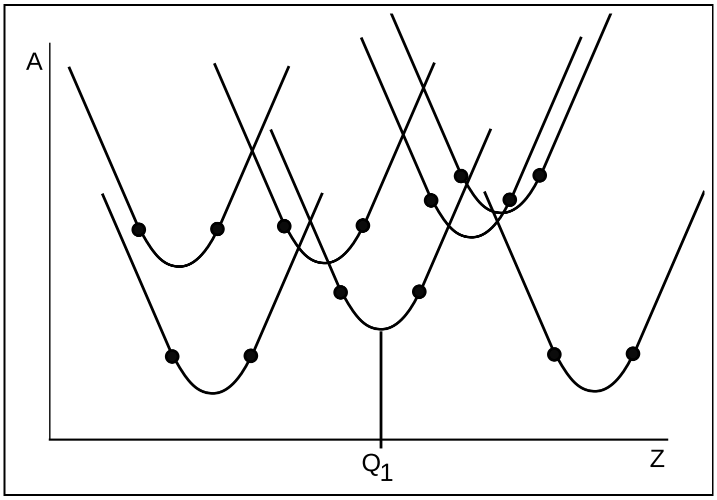

Figure 3.7: The median Q_1 of function minima.

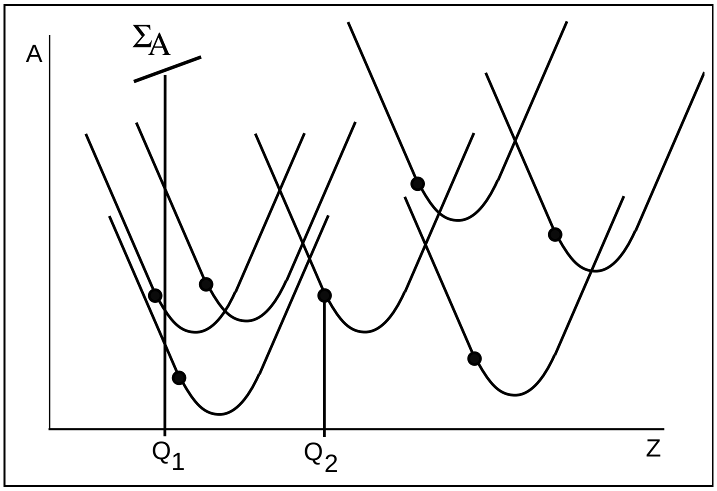

Figure 3.8: The median Q_2 of left inflection points.

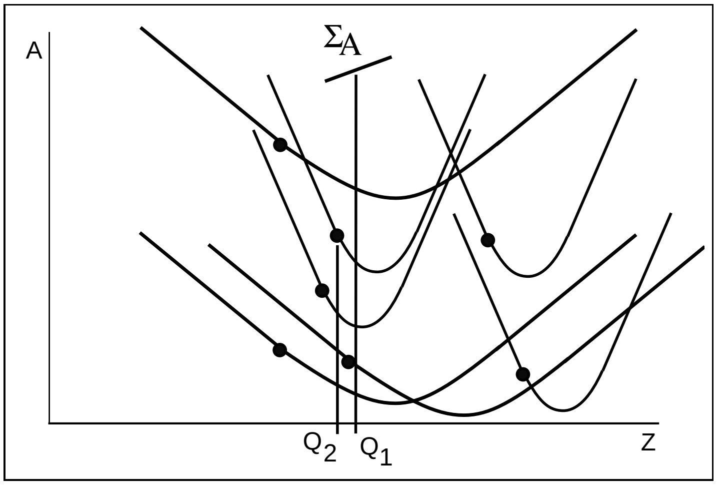

Figure 3.9: Q_2 to the left of Q_1.

The algorithm does $O(|R|)$ work in each iteration, and a constant fraction of R is removed each time. Thus the total time is $O(n)$, by a simple geometric series summation, given in [CLRS01].

Theorem 3.5. *The minimum area between two x-monotone chains, found over all vertical translations, can be computed in $O(n)$ time.*

Updating the aggregate function as we shift one of the chains along the x-axis appears to be non-trivial. It is no longer true that the optimal position must occur when vertices from each chain are aligned vertically. Also, when we make a small shift along the x-axis, not only do the two linear parts of each individual function change slope, but the center of symmetry of each function also may shift (Recall that these are functions of the z-coordinate). These changes depend on the slopes of our chains within each strip and are not difficult to compute on an individual basis. However understanding their aggregate effect is a different matter. To rephrase, each strip now has *three* "z-events" instead of one (the two boundaries between linear and quadratic forms, plus the center of symmetry). To make things worse, the z-events change position as a chain is shifted along θ. So if a tree is used to maintain the median, it will be necessary not only to insert/delete leaves but also to rearrange the order of leaves (to say the least).

3.4.4 Integer weights/heights

Here we discuss the cases where only certain pitches (heights) and/or weights are allowed.

If there are $O(1)$ height differences allowed, we can sort all critical points in $O(nm \log m)$, and sweep along each height difference horizontally, updating the area function in $O(1)$ time per critical point (i.e. $O(mn)$ per height difference), so the time complexity is dominated by the sorting step. Even in the simplest case, where we just wish to compute the minimum area while keeping z fixed, we do not know how to avoid sorting all critical positions.

If all weights are equal (i.e. we have evenly spaced sampling of melodies), then each median computation takes $O(m)$ time and there are $O(n)$ critical positions. Thus a brute force approach takes $O(nm)$ time. A direct implementation of our tree algorithm would take $O(nm \log m)$ time, since at each of the $O(n)$ critical positions we would have to update all $O(m)$ leaves of our tree. It is possible that this can be greatly improved.

Chapter 4

Reconfiguring Triangulations

4.1 Definitions and background

We now turn to the classic type of question posed in this thesis: *can object A be transformed to object B, using a defined set of operations?* The first polygonal object types considered in this context are planar *geometric triangulations.* Recall that a geometric *near-triangulation* consists of a set of points, pairs of which are joined by straight edges, so that no two edges intersect at their interiors and no more edges can be added. As a result, the convex hull of the point set is subdivided into triangular faces. The region outside of the hull is known as the *outer face.* If the outer face is incident to only three vertices (i.e. the hull contains only three vertices), then the object is plainly called a *triangulation.*

We allow (near)-triangulations to be reconfigured with two types of operations: *edge flips* and *point moves.* An edge flip on a triangulation is simply the deletion of an edge, followed by the insertion of another edge such that the resulting graph remains a triangulation. A point move is the modification of the coordinates of one vertex such that after the modification the graph remains a geometric triangulation.

Recall that it was Abellanas et al [ABG$^+$04], who introduced the *point move* in the context of triangulations, to allow reconfigurations between geometric triangulations on different point sets. They showed that this can be done with $O(n^2)$ edge flips and $O(n)$ point moves. For labeled triangulations, a quadratic number of each operation is used.

When restricted to edge flips, $\Omega(n^2)$ flips are sometimes necessary to transform one triangulation on a given point set to another one on the same point set [HNU99]. However, are $\Omega(n^2)$ edge flips required if one is also allowed to use point moves?

We give an algorithm which uses $O(n \log n)$ edge flips and point moves to transform any geometric near-triangulation on n points to any other geometric near-triangulation on n possibly different points, even if the points are labeled. It is important to realize that we do not restrict points to remain on a grid after each operation. Whether the same result may be achieved for grid points is still unknown. We also show that if we restrict our attention to geometric near-triangulations defined on a fixed point set of size n, i.e. the setting studied in [Law77], the problem is just as difficult even in the presence of point moves. Specifically, we show that if there exists an algorithm that transforms any near-triangulation on an n-point set to any other near-triangulation on the same point set using $O(n)$ point moves and edge flips, then this algorithm can be used to solve the more general problem of transforming any near-triangulation on one point set to any other near-triangulation on a possibly different point set with $O(n)$ operations. Finally, we show that with a more general point move, we can remove the extra log factor from our main result.

4.2 Some useful tools

From now on we assume that all triangulations and near-triangulations are geometric, and that the n vertices of any given triangulation are in general position. It is not difficult to see that $O(n)$ point moves can reconfigure a triangulation to this form. We begin with a few of the basic building blocks that will allow us to prove the main theorems.

Lemma 4.1. **[BCG$^+$03]** *A reconfiguration between two triangulations of the same point set that is in convex position can be done with* $O(n)$ *edge flips.*

Lemma 4.2. **[HNU99]** *Let* v_1*,* v_2 *and* v_3 *be three consecutive vertices on the outer face of a near-triangulation* T_1*. Let* C *be the path from* v_1 *to* v_3 *on the convex hull of all vertices but* v_2*. A near-triangulation* T_2 *containing all edges of* C *may be constructed from* T_1 *with* t *edge flips, where* t *is the number of edges initially intersecting* C *in* T_1*.*

Lemma 4.3. *Given a near-triangulation* T*, any vertex* $p \in T$ *with degree* $d > 3$ *that is inside the convex hull of the vertices of* T *can have its degree reduced to* 3 *with* $d-3$ *edge flips.*

Proof. Let P be the polygon that is the union of all triangles incident to p. P is a star-shaped polygon and p is in the kernel. By Meister's *two-ears theorem* [Mei75], if P has more than three vertices, then it has at least two disjoint ears[1]. At most one of them can contain p. Therefore p and one of the ears form a convex quadrilateral. We may flip the edge from p to the tip of the ear, effectively cutting the ear from P and reducing the number of vertices of P by one (see Figure 4.1). This process may be continued until P is reduced to a triangle that contains p as desired. □

Lemma 4.4. *Given a near-triangulation* T*, any vertex* $p \in T$ *with degree* 3 *that is inside the convex hull of the vertices of* T *can be moved to a new position in the triangulation along a straight path crossing* t *edges, using at most* $2t$ *edge flips and* $2t+1$ *point moves, assuming the path does not cross through any vertices.*

Proof. Suppose that p is joined by edges to vertices v_1, v_2 and v_3. Without loss of generality, let edge v_2v_3 intersect the path that p must follow, and let this path continue into triangle $v_2v_3v_4$, as shown in Figure 4.2.

[1] A triangle, defined by three consecutive vertices of a polygon, is an ear if it is empty and the vertices form a convex angle. The second vertex is the *tip* of the ear.

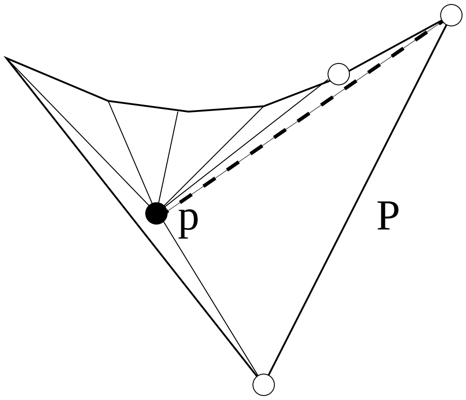

Figure 4.1: Polygon P is visible from vertex p, which is inside one of the two ears of P. The empty ear is labelled with white vertices. The edge that may be flipped is dashed.

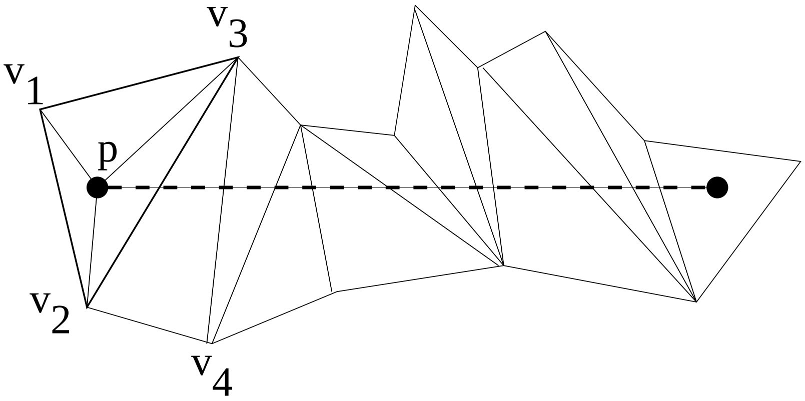

Figure 4.2: A vertex p and a straight path that it must move along (dashed). p can pass through any edge with two edge flips.

Clearly p can be moved anywhere within triangle $v_1v_2v_3$ without the need of any edge flips. Then it can be moved along its path, as close to edge v_2v_3 as necessary, so that the quadrilateral $pv_2v_3v_4$ becomes convex. This allows edge v_2v_3 to be flipped into edge pv_4. Now p may continue along its path. As soon as it enters $v_2v_3v_4$, edge pv_1 may be flipped into v_2v_3. Now, with two edge flips and two point moves, p has crossed through the first edge intersecting its path, and still has degree 3. By the same argument, p may traverse its entire path with two edge flips and two point moves for each intersecting edge. One additional point move is required in the last triangle. Note that only three edges in the original and final triangulations will be different. □

Lemmata 4.3 and 4.4 imply the following result:

Lemma 4.5. *Given a near-triangulation T, any vertex in the interior of the convex hull of the vertices of T with degree d can be moved to a new position in the triangulation along a path crossing t edges, using $O(d+t)$ edge flips and point moves.*

Lemma 4.6. *An edge can be constructed between a convex hull vertex and any other vertex in a triangulation using $O(n)$ edge flips, with the aid of one moving point that is moved $O(n)$ times.*

Proof. Let v_1 be the hull vertex. First suppose that the second vertex is an interior point. Then it will play the role of the moving point, and we will label it p. We can move p directly towards v_1, until it is located within a triangle that has v_1 as a vertex. Now v_1 and p must be joined with an edge. Next we move p back along the same line to its original position, always maintaining edge v_1p. To do this, we consider the set of triangles that intersect p's path, as in lemma 4.4. Vertex p can always enter a triangle intersecting the path back to its original location. The difference is that once it has crossed an intersecting edge, we do not restore the edge. This means that p will accumulate edge degree. An issue that needs to be taken care of is that of

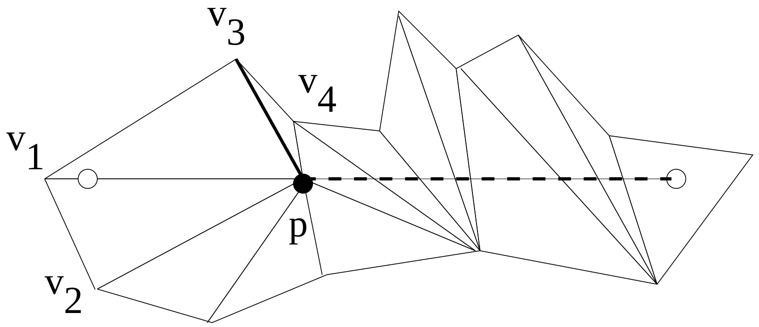

Figure 4.3: Maintaining a triangulation while extending edge v_1p: p has moved from a position close to v_1 (shown white), and still has to traverse the dashed segment to its original position. Edge pv_3 causes a problem if p is to continue.

maintaining a triangulation when p is about to lose visibility to another vertex. This occurs when one of its incident edges is about to overlap with another edge in the triangulation, as shown in Figure 4.3.

Suppose that edge pv_3 is about to overlap with edge v_3v_4. v_3 and v_4 cannot be on opposite sides of the remaining path that p must traverse, otherwise v_3v_4 may be flipped. Vertex p must share an edge with v_4 in this configuration. Vertices p and v_3 are also part of another triangle, along with some vertex v^* which may be anywhere on the path from v_1 to v_3. These two triangles must form a convex quadrilateral $pv^*v_3v_4$, otherwise p would have already lost visibility to v^*. Thus pv_3 may be flipped into v_4v^*, which means that v_3 is removed from the polygon that intersects p's path. The result is that when p reaches its original position, it leaves a *fan*[2] behind it, which includes edge v_1p. Overall one edge flip is used when p enters a new triangle, and at most one flip is used for every edge that attaches to p.

If both vertices of the edge that we wish to construct are on the hull, then we can take any point p within the hull and move it close to v_1 and onto the segment between the two hull vertices. p can then move along this segment to the second hull vertex until it is connected to both. At this moment, p may be perturbed so that the

[2] A fan is a star-shaped polygon with a vertex in its kernel.

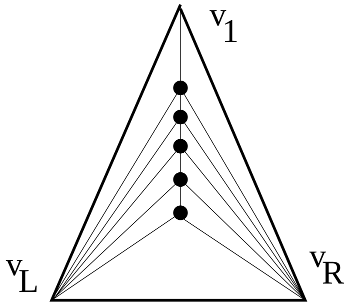

Figure 4.4: The canonical configuration used for triangulations.

three vertices form a triangle. This triangle might contain other edges incident to p. Lemma 4.2 implies that these edges may be removed so that the desired edge can be constructed with $O(n)$ edge flips. □

4.3 A divide and conquer algorithm

Here we give an algorithm which uses $O(n \log n)$ total operations to perform any reconfiguration.

4.3.1 Triangulations

With the basic building blocks in place, we now prove one of our main results.

Theorem 4.7. *With $O(n \log n)$ edge flips and point moves, we can transform any geometric triangulation on n points to any other geometric triangulation on n possibly different points.*

Proof. We transform one triangulation to another via a canonical configuration. As shown in Figure 4.4, the interior vertices form a *backbone* (i.e. their induced subgraph is a path). The top of the backbone is joined to the topmost hull vertex v_1, and all interior vertices are joined to the other two hull vertices, v_L and v_R.

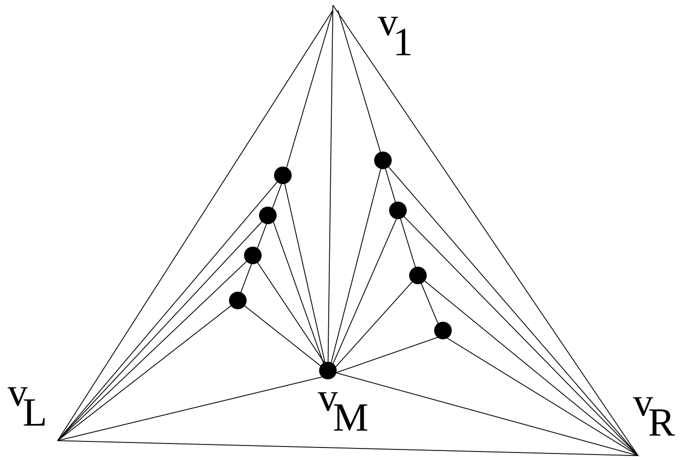

Figure 4.5: The configuration of a triangulation prior to merging the backbones on each side of the median vertex v_M.

The canonical configuration is constructed in a divide-and-conquer manner. We perform a radial sweep from v_1, to find the median vertex interior to the convex hull, v_M. After constructing edge v_1v_M we move v_M directly away from v_1 towards the base v_Lv_R, maintaining v_1v_M until triangle $v_Mv_Lv_R$ contains no interior points. By Lemma 4.6, we use $O(n)$ operations to accomplish this. Now, we transform $v_1v_Mv_L$ and $v_1v_Mv_R$ into backbone configurations by induction since they are smaller instances of the same problem. The resulting configuration is shown in Figure 4.5.

We now show that the two sides may be merged using $O(n)$ operations. As shown in Figure 4.6a, we first move the lowest vertex of a backbone into a position that is close to the base and is along the extension of edge v_1v_M. This requires one edge flip. The vertices on the left/right backbones are processed in ascending order, and are always moved just above the previous processed vertex, as shown in Figure 4.6b. Each vertex will require two point moves and one edge flip. Thus $v_1v_Lv_R$ is reconfigured into canonical form, and by a simple recurrence the number of edge flips and point moves used is $O(n \log n)$. It is trivial to move a canonical triangulation to specific coordinates using n point moves. Thus the transformation between any two triangulations may be completed. □

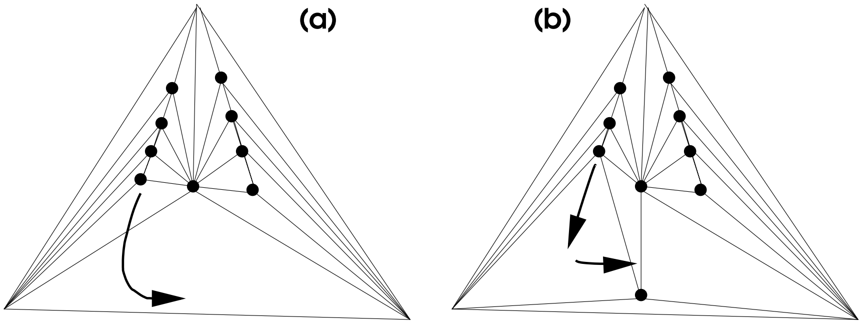

Figure 4.6: Merging two backbones into one.

4.3.2 Near-triangulations

If the initial graph is a near-triangulation, we assume that the outer face is a convex polygon. We also assume that the initial and target triangulations have the same number of hull vertices. This is necessary in order for both to have the same number of edges. Since the outer face is not a triangle, Theorem 4.7 does not directly apply. Some care must be taken to handle a non-triangular outer face. We outline the details below.

Theorem 4.8. *With $O(n \log n)$ edge flips and point moves, we can transform any geometric near-triangulation on n points to any other geometric near-triangulation on n possibly different points.*

Proof. As in the case with triangulations, we transform one near-triangulation to another via a canonical configuration. In the primary canonical configuration, shown in Figure 4.7, one chosen hull vertex (v_1) is joined by chords to all other hull vertices. Thus v_1 is in the kernel of a convex fan. Every triangle incident to a hull edge, except for one, is empty. All interior vertices, located in the non-empty triangle T, are in the canonical configuration of a triangulation. Once this configuration is achieved, all vertices can easily be placed at specific coordinates, so that the transformation between two-near triangulations can be completed. This will be described further on.

We first construct all edges of the top-level fan configuration, leaving interior

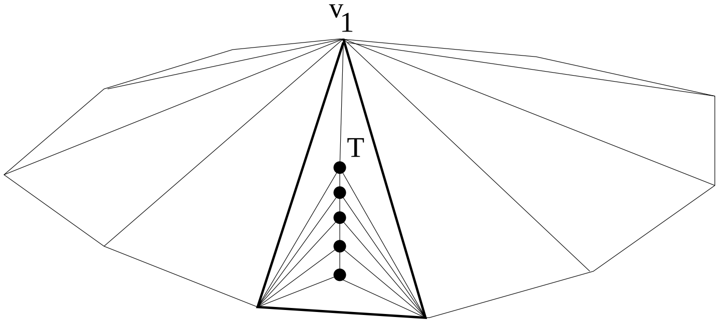

Figure 4.7: The primary canonical configuration used for near-triangulations.

vertices in their original positions. Then within each triangle of the fan, we rearrange the interior vertices into a canonical triangulation. Finally, we merge all triangles of the fan, so that all interior points move to a single triangle and are in canonical form.

To construct the fan chords, we always divide the problem into two roughly equal parts. We begin by constructing two chords as follows: perform a radial sweep from v_1 to successive hull vertices v_i $\{2 \leq i \leq n-1\}$, always keeping fewer than $\frac{n}{2}$ vertices in the swept region. Let v_j be the last hull vertex for which this holds. Construct chords v_1v_j and v_1v_{j+1}. The unswept region not including triangle $v_1v_jv_{j+1}$ contains fewer than $\frac{n}{2}$ vertices. The swept region contains fewer than $\frac{n}{2}$ vertices. Triangle $v_1v_jv_{j+1}$ may contain an arbitrary number of vertices, but this is not a sub-problem (we will not look at this region again during the construction of the fan). Now we can continue a new sweep on each side of $v_1v_jv_{j+1}$. Construction of the two chords could take $O(n)$ edge flips and point moves, as described in Lemma 4.6. However the even split of the sub-problems ensures that the total number of operations is $O(n \log n)$.

Each fan triangle $v_1v_iv_{i+1}$, containing k_i interior points, can be reconfigured into a backbone structure with $O(k_i \log k_i)$ operations, by Theorem 4.7. Thus the total number of edge flips and point moves used to reconfigure all triangles of the fan into backbone structures is $O(n \log n)$.

Now we are left only with the task of merging the fan triangles so that only one of them will contain all interior points. To do this, we pair up consecutive triangles,

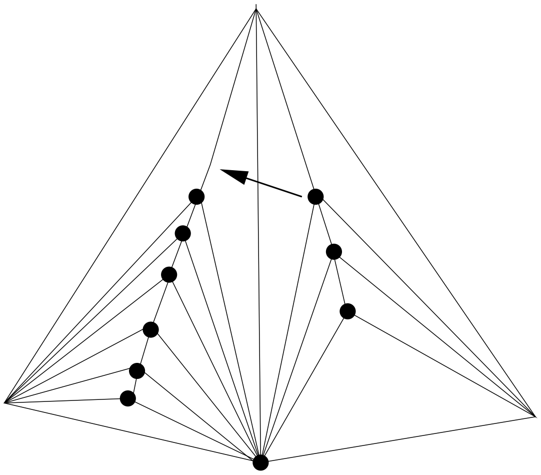

Figure 4.8: Merging two adjacent fan triangles.

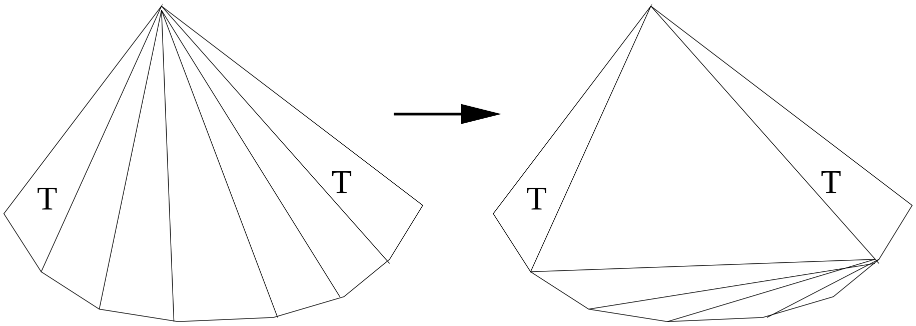

Figure 4.9: Handling multiple adjacent empty fan triangles. Triangles marked (T) contain triangulations.

merge them, and continue recursively: We can add k_i interior points of a canonical triangulation to an adjacent canonical triangulation using $O(k_i)$ edge flips and point moves. The k_i points are processed in descending order and are always added to the top of the adjacent triangulation, as shown in Figure 4.8.

Thus we obtain one triangle in canonical form next to an empty triangle. It is just as easy to merge two canonical triangles separated by an empty triangle. If we ever encounter two or more adjacent empty fan triangles, we may use Lemma 4.1 to reconfigure them so that they will not affect the fan-merging process (see Fig 4.9).

By the above arguments, once we select the triangle that is to finally contain all

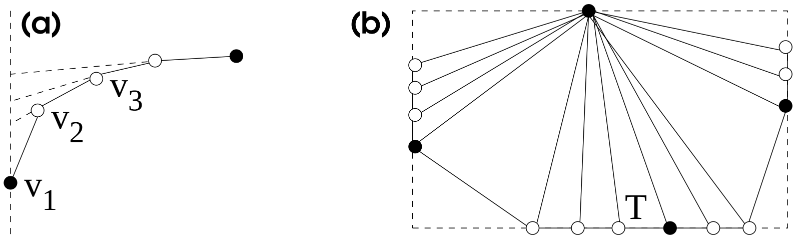

Figure 4.10: (a) Moving a vertex onto the bounding rectangle. (b) All hull vertices on the bounding rectangle.

of the interior points (the median triangle is a good choice), we can iteratively merge its neighbouring triangles onto it using a total of $O(n)$ edge flips and point moves.

Finally we are left with a single triangle containing all interior points in canonical form. On either side, we may have an arbitrary triangulation (resulting from handling multiple adjacent empty fan triangles), but the vertices will be in convex position. By Lemma 4.1 they may be moved to our desired configuration using $O(n)$ edge flips.

We must still show that this primary canonical configuration can be moved to specific coordinates. This can be done with $O(n)$ point moves: First we move all vertices onto the bounding rectangle, by processing each of the hull paths between extreme vertices X_{max}, X_{min}, Y_{max} and Y_{min} separately. Let the path from X_{min} to Y_{max} contain vertices $X_{min} = v_1, \ldots, v_k = Y_{max}$. Vertex v_2 can be moved directly away from v_3 until edge v_1v_2 becomes vertical, as shown in Figure 4.10a. Similarly, vertices $v_3, \ldots, v_{k-1}$ may be moved to this vertical line through X_{min}. By performing similar motions for the other paths, we obtain a configuration as the one in Figure 4.10b. In each case one point move suffices, except for the hull vertices belonging to the triangle that contains the interior points. To move these two vertices, we have to displace the interior points, but one point move per interior point suffices.

Now it is trivial to move all hull vertices except for Y_{max} along the edges of the bounding rectangle so that they reach the bottom edge. This allows the top vertex to move to any chosen coordinate above the bottom edge. All remaining vertices may be shifted horizontally to any position. Finally, by moving them again along the boundary of their new bounding rectangle, they can be positioned on the two vertical sides, which now allows them to be moved to any position vertically. Thus the reconfiguration may be achieved within the bounding box of the source and target triangulations. □

4.3.3 Reconfiguring triangulations on a fixed point set

If two triangulations have the same point set, the problem is no easier than the general problem. Suppose that there exists an algorithm that can transform a triangulation T_1 on a given n-point set to a triangulation T_2 on the same point set using $F_n = o(n \log n)$ edge flips and point moves[3] . Then this algorithm can be used to transform a triangulation on one point set to any other triangulation on a possibly different point set with $F_n + O(n)$ edge flips and point moves. This argument is summarized in Fig 4.11. Let Fig 4.11(a) be the input triangulation. With F_n flips and moves, move to the triangulation in Fig 4.11(b) where every interior vertex is adjacent to the lower left vertex v_ℓ of the outer face.

Now consider the triangulated polygon, P, that consists of edges not adjacent to v_ℓ. Notice that if we perform a radial sweep from v_ℓ, the boundary of P is monotonic. At least two of the triangles in P are disjoint ears, which means there must exist an ear tip that is an interior vertex and is also joined to v_ℓ by an edge in the original triangulation. We may move this point directly towards v_ℓ and cut the ear from P. This still leaves a monotone polygon P'. By continuously locating such ears, and moving them to a predefined convex position, we can obtain the configuration illustrated in Figure 4.11c. The monotonicity of P (and its descendants) and the convexity of the final configuration of interior points guarantee that no edge crossings will occur. This process requires a linear number of point moves.

[3] The upper bound of a function F_n is $o(f_n)$ if the limit of $\frac{F_n}{f_n}$ approaches infinity. In other words, f_n is strictly more than just a constant multiple of F_n.

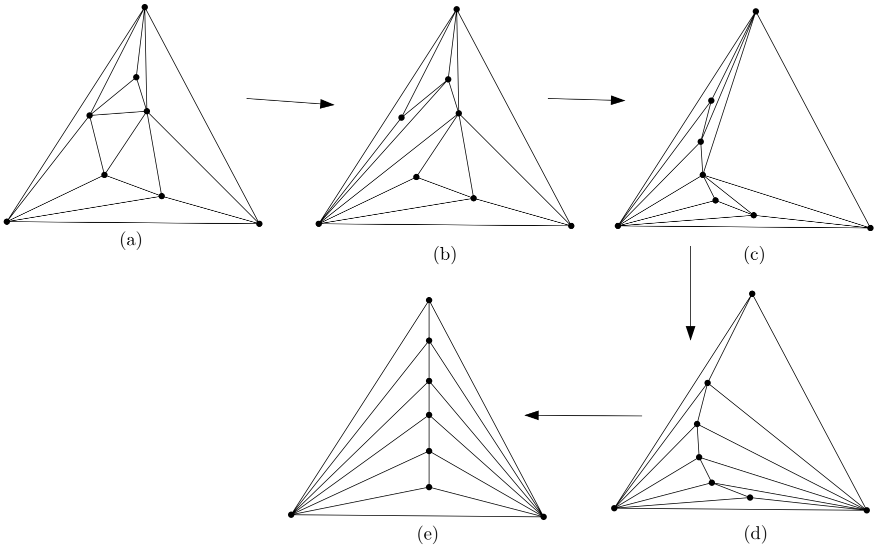

Figure 4.11: Problem on fixed point set is not easier.

Next, by Lemma 4.1, we can use $O(n)$ edge flips to obtain the triangulation where the lower right vertex of the outer face is adjacent to every vertex, as illustrated in Figure 4.11d. From here, it is trivial to move to the canonical configuration.

We conclude with the following:

Theorem 4.9. *If an algorithm exists that can reconfigure between any two geometric triangulations of the same point set with $o(n \log n)$ edge flips and point moves, then we can also transform any geometric triangulation on n points to any other geometric triangulation on n different points with $o(n \log n)$ flips and moves.*

It is tempting to try to find a fast algorithm that will construct a monotone path, as illustrated in the transition from Figure 4.11a to Figure 4.11b. Consider the polygon that is the union of all triangles incident to the lower left vertex of Figure 4.11b. By continuously cutting ears of this polygon, we may get to a triangulation that is *similar* to that of Figure 4.11a, using $O(n)$ edge flips. The similarity is that all neighbours of the lower left vertex will be in convex position. However, we have little control over the resulting positions of the remaining edges if we use only $O(n)$ operations. It is possible to create triangulations for which the reversal of this ear-cutting technique is not possible. In fact, Figure 4.11c serves as an example, if we add a few more vertices inside the large triangle. In this figure none of the edges directly visible from the lower left vertex can be flipped, so there is no obvious way to achieve a monotone path by using operations only in the neighbourhood of v_ℓ.

4.4 Remarks

An interesting result by Aichholzer et al [AAC$^+$96] is that there is a matching between the edges of any two triangulations of a point set, such that an edge of one triangulation is matched either to the identical edge in the other triangulation or to an edge that crosses it. This result might prove to be useful for reconfigurations.

The results of this chapter also hold for labelled triangulations. The merging step for triangulations can be easily adapted so that two backbones, sorted by their labels, can be merged into one sorted backbone in a procedure that resembles merge-sort. For near-triangulations, merging adjacent backbones as in Figure 4.8 results in one backbone inverting its sorted order. This must be taken into consideration. Note that in our algorithms a point move can be visualized as a continuous motion of a vertex. In the case of labelled triangulations, if it is necessary to change points on the convex hull, “continuous” point moves are inadequate. Instead, the coordinates of a hull vertex v_h can be modified instantaneously to a position within the hull, as long as an interior point exists close enough to v_h.

Finally, consider a more powerful point move, where we can delete an interior vertex of degree three (and its incident edges), and create a new vertex of degree three elsewhere. With this type of move we can reconfigure triangulations using $O(n)$ operations. We select a triangle incident to a hull edge and create a backbone inside. This is done by repeatedly selecting a vertex of constant degree from outside the triangle, reducing its degree to three, and moving it to the lower end of the backbone.

Chapter 5

Locked Hexagonal Linkages

5.1 Background

We now turn to one of the most intuitive classes of polygonal Structures: three-dimensional polygons. The method of reconfiguration is also one of the easiest to visualize: basically anything is allowed, as long as no edges cross or deform. This is the *universal* model of motion for polygonal linkages.

Consider a hexagon in $\mathbb{R}^3$ with fixed edge lengths[1] $\ell_1, \ldots, \ell_6$, and for which we allow universal joint motions but no edge crossings. In other words, we allow the vertices $v_1, \ldots, v_6$ of the hexagon to move freely as long as the edges remain straight, and do not cross or change length. Following the notation of Cantarella and Johnston [CJ98], we denote the space of this hexagon as $Pol_6(\ell_1, \ldots, \ell_6)$. The general question that we study here is: *how many connected components can Pol_6 have, for a suitable choice of $\ell_1, \ldots, \ell_6$?* It is then natural to ask whether each connected component of Pol_6 corresponds to a separate knot type, or if some knot type (say the unknot, or trivial knot) exists in separate components of space. Millet [Mil94] showed that for regular hexagonal unknots there is only one embedding class. The question remained open

[1] We use ℓ_i to denote both the edge and its length.

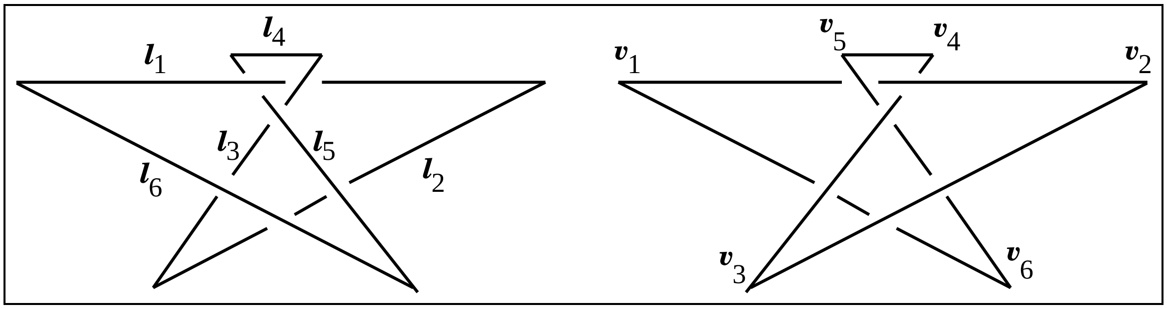

Figure 5.1: Unknot classes 2 and 3, by Cantarella and Johnston. (not drawn to scale)

until Cantarella and Johnston proved that there exist three connected components in $Pol_6(100, 63.5, 22.7, 5.6, 22.7, 63.5)$, each belonging to the unknot[2]. The first component includes the planar convex embedding. Recall from Chapter 2 that all convex polygons are in the same component. The remaining two classes, which are left and right hand versions of the same polygon, were shown to be "stuck", in the sense that they cannot be reconfigured into planar convex embeddings. These classes are shown in Figure 5.1.

Cantarella and Johnston suspected that these three classes were the only ones for the unknot in Pol_6, but Toussaint [Tou01] showed that there exist two more classes, for $Pol_6(20, 13, 4, 1, 4, 13)$. The two new classes are left and right hand versions of each other, and are shown in Figure 5.2. In the following section, we show that four more classes can exist for appropriate edge lengths.

5.2 A hexagonal unknot with nine embedding classes

We consider the space $Pol_6(1, 1, 0.55, \epsilon, 0.55 - \epsilon, 1)$, where $\epsilon \leq 0.01$ (i.e. sufficiently small). Here, we show that this space has at least nine connected components corresponding to the unknot. In other words there are at least nine embedding classes of the unknot for our given edge lengths.

[2] The numbers given are approximate.

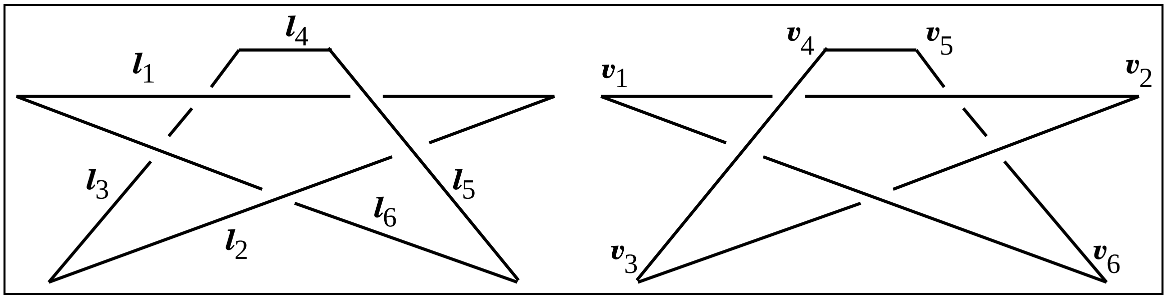

Figure 5.2: Unknot classes 4 and 5, by Toussaint. (not drawn to scale)

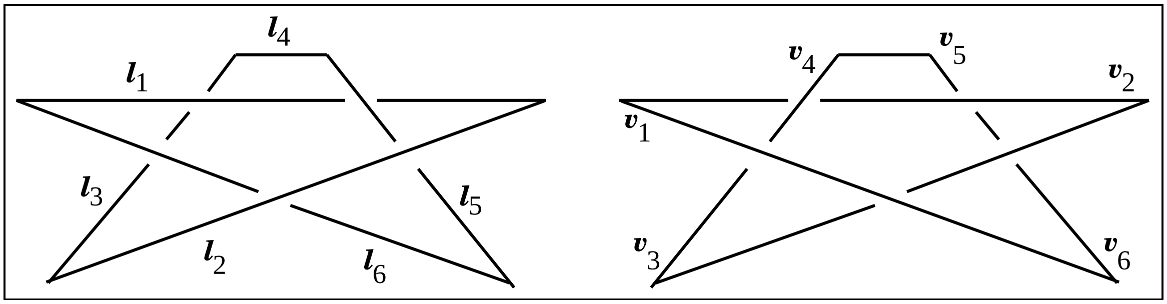

Figure 5.3: New unknot classes 6 and 7. (not drawn to scale)

The first class contains all planar convex embeddings. The next four are similar in shape to those of Cantarella and Johnston and those of Toussaint. However, since we are using different edge lengths it is necessary to verify that our classes of hexagons are still in different connected components of space. The four new classes are shown in Figures 5.3 and 5.4.

Cantarella and Johnston showed that a sufficient condition for their classes to be in different connected components than the convex class is that $(\ell_3 + \ell_4)^2 < \ell_1\ell_2 - (\ell_1)^2/2$. Our chosen edge lengths satisfy this condition, so we can say that the proof of Cantarella and Johnston holds for our hexagon. Of course, it is not enough just to show that every class is in a different connected component than that containing the convex class.

Unfortunately, we can not do the same as above and borrow the proof for Tou-

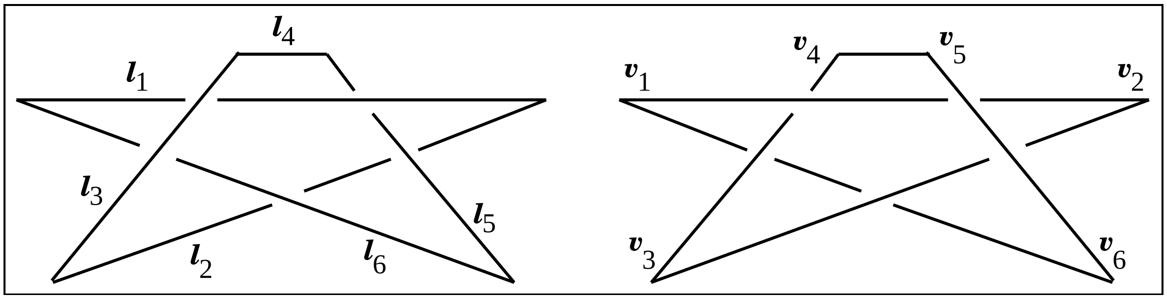

Figure 5.4: New unknot classes 8 and 9. (not drawn to scale)

ssaint's classes. A sufficient (but not necessary) condition used in this case is that $\ell_3 + \ell_4 + \ell_5 < min\{\ell_2, \ell_6\}$. Our hexagon is modified enough that this condition no longer holds. It seems that we cannot avoid such a modification, in order to create the new classes.

Even though Toussaint's proof cannot be used here, the intuition is quite the same. For each class, there are a couple of motions that must be made in order to convexify the hexagon. We will show that such motions are impossible to make, and conclude that it is impossible not only to convexify a hexagon belonging to each class but also to reconfigure between the non-convex classes.

Let us now fix a coordinate system in order to view all possible motions. Let ℓ_1 be fixed in the xy plane, and ℓ_6 be constrained to the plane. Specifically, v_1 is at $(0, 0, 0)$, v_2 is at $(1, 0, 0)$, and v_6 is at $(v_{6x}, v_{6y}, 0)$ where $v_{6x}^2 + v_{6y}^2 = 1$. Our view will be along the normal of the plane. Thus vertex v_6 may move only in a circle about v_1. We choose to focus on Class 9, as shown on the right of Figure 5.4. Intuitively we can see that to convexify this hexagon we would have to pass v_3 over ℓ_6 and into the plane, or we would have to pass ℓ_5 over v_2 so that v_5 could be placed in the plane.

We now proceed to describe certain constraints in the configuration of our polygon. The polygon shown in Figure 5.4 possesses certain properties, listed here:

1. $v_{5z} > 0$, and $v_{4z} < 0$. Thus we can define the point $P = P_{(x,y)} = (P_x, P_y)$ to be the intersection of ℓ_4 with the plane. This also means that v_4 and v_5 are at a distance of at most ϵ from the plane.

2. $P_y > 0$, and $\epsilon < P_x < 1 - \epsilon$.

3. $v_{3z} > 0$.

We will show that, while reconfiguring the polygon, it is impossible to change any single property without causing an intersection or contradicting the validity of at least one other. We will also show that we cannot change more than one of these properties simultaneously. This implies that the properties are always true. Note that *changing* a property is done via a continuous motion of the polygon. Thus for example, property 1 changes the moment that $v_{5z} = 0$ and/or $v_{4z} = 0$. In case we attempt to set both of these vertices into the plane simultaneously, the point P will be defined to be the last unique point of intersection of ℓ_4 with the plane.

First we attempt to change only the third property. Consider the case where $v_{3y} < 0$. v_3 can be placed in the plane only if the angle at v_2 or the angle at v_1 opens to more than $\pi/3$. This would mean that the distance from v_3 (or v_6 respectively) to ℓ_1 becomes greater than $\frac{\sqrt{3}}{2}$. The distance to P is at least as great, since $P_y > 0$. However, the distance from v_3 (v_6) to P can be at most the sum of the lengths $\ell_3 + \ell_4$ (or $\ell_5 + \ell_4$ respectively). These sums are less than $\frac{\sqrt{3}}{2}$. Thus, by contradiction, we conclude that v_3 cannot be placed in the plane if $v_{3y} < 0$. Allowing properties 1 and/or 2 to change simultaneously with property 3 does not affect the arguments given above. At the "critical" moment that we attempt to place v_3 into the plane, we still have $P_y \geq 0$, for example.

It remains to be seen if we can change only the third property, when $v_{3y} > 0$. Even reaching a configuration where $v_{3y} > 0$ requires a subtle motion. It can be done by bringing P sufficiently close to ℓ_1, which allows us to obtain $v_{4y} < 0$. Then ℓ_2

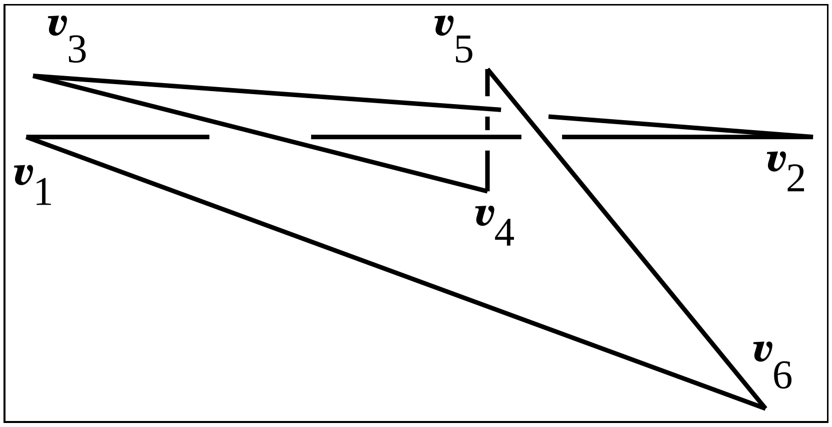

Figure 5.5: Alternate configuration of Class 9. (not drawn to scale)

can move in a clockwise motion as viewed from the normal to the plane. The final configuration is shown in Figure 5.5. During the motion described above, we have $0 \leq v_{3x} \leq 1$. Thus the x-coordinates of both endpoints of ℓ_3 are within this range. This means that if ℓ_3 is to avoid intersection with ℓ_1 it is in fact necessary to have $v_{4y} < 0$. Keeping in mind that $v_{4z} < 0$ (property 1), we see that it is impossible to place v_3 in the plane without causing an intersection between ℓ_3 and ℓ_1. We conclude that v_3 always remains above the plane, as long as properties 1 and 2 are true. Again, allowing properties 1 and/or 2 to change at the same time (i.e. allowing $P_x = \epsilon$ *and/or* $v_{4z} = 0$) doesn't affect our arguments. Thus property 3 cannot change, either on its own, or at the same time as properties 1-2.

We now focus on the second property: To obtain $P_y < 0$, we must *first* have $P_x < 0$ or $P_x > 1$. This is because of the position of ℓ_1. So we would be requiring that $v_{4x} < \epsilon$ or $v_{4x} > 1 - \epsilon$. The same holds for v_{5x}. In other words, v_4 and v_5 would have to approach to within a distance ϵ from the halfline $x = 0, y > 0, z = 0$ or the halfline $x = 1, y > 0, z = 0$. By examining the triangle $v_1v_5v_6$ (which can be made to approach arbitrarily close to the plane if ϵ is small enough) and using elementary trigonometry, we can see that the former case is impossible. On the other hand, for v_4 to approach the halfline $x = 1, y > 0, z = 0$, we would need the angle $v_1v_2v_3$ to reach

$\pi/3$. As shown previously, this cannot be done when $v_{3y} < 0$. We also established that if $v_{3y} > 0$, as shown in Figure 5.5, then $v_{4y} < 0$. This means that the angle $v_1v_2v_3$ cannot open to $\pi/3$, by the same logic used when dealing with changing property 3. Thus we must always have $0 < P_x < 1$ and $P_y > 0$. Once again, the argument given is not affected by allowing property 1 to change simultaneously.

Finally, we can examine the first property: Given property 2, we cannot place v_5 into the plane because this would cause an intersection between the polygonal arc Pv_5v_6 and ℓ_1. Also, we cannot place v_4 into the plane because the arc v_3v_4P would intersect ℓ_1. More specifically, if $v_{3y} \geq 0$, bringing v_4 to the plane would cause an intersection between ℓ_4 and ℓ_1, since $P_y > 0$. If $v_{3y} < 0$, the intersection would be between ℓ_1 and either ℓ_3 or ℓ_4 depending on the value of v_{4y}. We conclude that v_4 (v_5) must always be below (above) the plane.

Having established property 1, we know that Class 9 cannot be reconfigured to classes 2,5,7,8. Now consider Class 6: ℓ_2 passes above the polygonal arc Pv_5v_6, whereas in Class 9 it passes beneath the arc. In both cases, ℓ_2 is above the plane. We know that starting from Class 9 we cannot position v_3 into the plane. This implies that if we are to reconfigure between these classes, v_3 must pass under ℓ_5. This cannot be done, due to the angle restrictions mentioned while handling property 1. In fact, if this were possible, then v_3 could be placed in the plane. The same applies for Class 3. Finally, the only difference between classes 4 and 9 is in the crossing of ℓ_3 and ℓ_6, as we view along the normal to the plane. To change this crossing, the point Q, defined earlier, would have to move to an intermediate position such that $Q_y < v_{6y}$. This involves passing v_3 under ℓ_5, which cannot be done, as we have just seen.

We can now say that Class 9 cannot be reconfigured to any of the other classes shown. If we ignore the small difference in the lengths of ℓ_3 and ℓ_5, which does not play a role in the proofs given above, then the four new classes are either left/right hand versions or rigid transformations of each other. Thus the arguments given for

Class 9 hold for classes 6–8. As for our modified examples of Toussaint's classes, the proofs are similar and need not be repeated. This leads us to the following theorem:

Theorem 5.1. *For suitable choices of edge lengths, there are at least nine connected components belonging to the unknot in* Pol_6.

5.3 Remarks

Although we show that there are at least four more embedding classes for Pol_6, we note that our examples are not very "stable", in the sense that the slightest change in the length of either ℓ_2 or ℓ_6 would result in the loss of four classes: those of Toussaint, and two of the four new ones. If we made both lengths larger, then we would only have the three classes of Cantarella and Johnston. Therefore an open problem is to find suitable edge lengths so that Pol_6 has more than five "stable" embedding classes.

As was mentioned by Cantarella and Johnston, these results can be extended to Pol_n, by replacing the shortest edge with a chain. Alternatively, a link or a knot can be placed to obtain similar results.

It was tempting to see whether we could modify our classes, by changing some crossings, to obtain more than four components corresponding to the trefoil knot. The examples of Cantarella and Johnston cannot be transformed into trefoils, and the remaining six non-planar unknots produce only four trefoil classes, due to symmetry conditions.

Chapter 6

Dihedral Motions

6.1 Definitions and background

We have seen that polygons and chains in $\mathbb{R}^3$ can "lock" even with the least restrictive model of motion. Here, we continue to deal with polygonal chains, but we will focus on the class of chains that have all edge lengths equal. This restriction makes it difficult to find chains that are locked using universal motions (if they even exist). One might say that unit-length chains are more flexible. However, the method of reconfiguration considered in this chapter is more restrictive.

Here, in addition to edge lengths, the angles between successive edges also cannot change. Informally, to perform a *dihedral motion* we fix one edge in space and rigidly rotate one of the two remaining subchains, using the fixed edge as an axis of rotation. As shown in Fig 6.1, the fixed edge forms a plane with each of its two neighbours. A dihedral rotation causes the angle between these planes to vary. This model is essentially the "ball and stick" model for molecules, used in introductory chemistry classes. The general dihedral chain reconfiguration problem is to determine whether two configurations of a three-dimensional chain are connected via a series of dihedral motions that preserve simplicity.

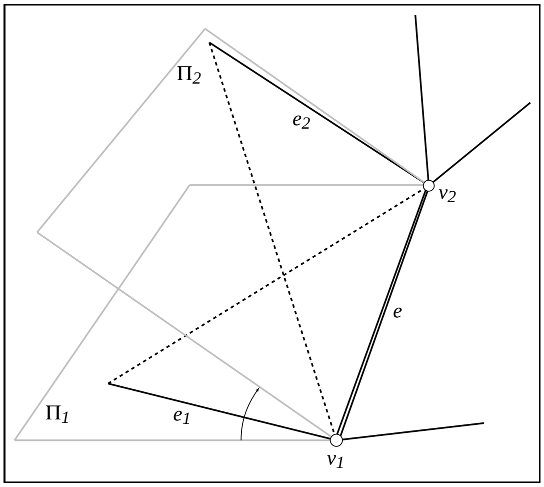

Figure 6.1: A local dihedral motion. e_1 and e are fixed in plane Π_1, while e_2 (in plane Π_2) rotates about e.

A major open problem is to determine whether any two *planar* configurations of a chain are "connected" by a series of dihedral motions, while maintaining simplicity at all times. If this is true for two chains, we say that they are "flat-state connected". This problem, which is now known as the "flat-to-flat" problem, was left open after an extensive study of dihedral motions by Soss [Sos01]. Planar configurations may be useful as an intermediate (canonical) form during the reconfiguration of three-dimensional chains.

Recent results have only been obtained by placing additional restrictions on angles and/or edge lengths. Some of our results that have appeared at *CCCG* and *ISAAC* (2002) [ADM+02, ADD+02] are summarized in Table 6.1.

Note a particular result on the flat-state connectivity of unit length chains for which all angles are in the range $(60°, 90°]$. The following section extends this result to the range $(60°, 150°)$.

Table 6.1: Summary of results on dihedral reconfigurations. The '—' means no restriction of the type indicated in the column heading. Entries marked '?' are open problems. Unless indicated otherwise, results are from [ADD+02].

Constraints on Fixed-Angle Linkage				Flat-state
Connectivity	Angles	Lengths	Motions	connectivity
Open chain	—	—	—	?
	has a monotone state		—	?
	non-acute	—	—	Connected [ADM+02]
	equal acute	—	—	Connected [ADM+02]
	each in $(60°, 90°]$	unit	—	Connected [ADM+02]
	—	—	180° edge spins	Disconnected
	orthogonal	—	180° edge spins	Connected
Set of chains, each pinned at one end	orthogonal	—	—	Connected
	orthogonal	—	partially rigid	Disconnected
Polygon	—	—	—	?
	nonacute	—	—	?
	orthogonal	—	—	?
	orthogonal	unit	—	Connected
Tree	—	—	—	?
	orthogonal	—	—	?
	orthogonal	—	partially rigid	Disconnected
Graph	orthogonal	—	—	Disconnected

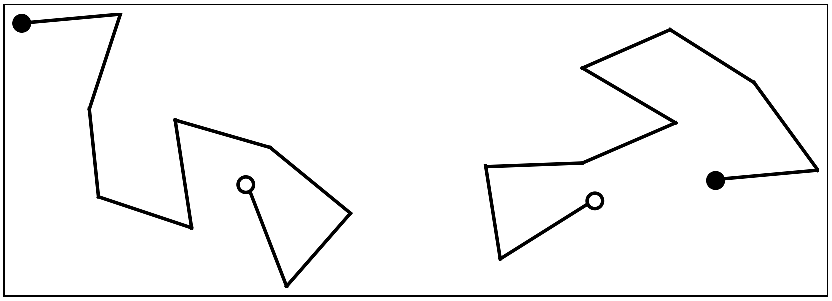

Figure 6.2: Two configurations of a unit-length chain with angles in the range $(60°, 150°)$.

6.2 Unit length chains

In this section we prove that unit length chains with all dihedral angles in the range $(60°, 150°)$ are flat-state connected. Two configurations of such a chain are shown in Figure 6.2.

We use a *canonical configuration* in a plane perpendicular to the original, as an intermediate between the two configurations. The definition of the canonical form is the following: the first edge v_1v_2 of the given chain must point up[1] . From there, each successive edge v_iv_{i+1} is placed so that v_{i+1} reaches a position with maximum height (without interfering with edges already fixed in place). This definition creates a unique chain in a vertical plane except for the case that an edge is vertical. In this case we can arbitrarily choose a direction for its successor, according to any rule that we like (for example, always choose a left turn).

Figure 6.3 shows how we would build the canonical configuration of the linkage in Figure 6.2. Alternate positions of each edge are shown dashed.

[1] We say that an edge v_iv_{i+1} *points down* if v_{i+1} is strictly lower than v_i. Otherwise the edge *points up*. Pointing left and right are defined similarly, with vertical edges symbolically defined to be pointing right.

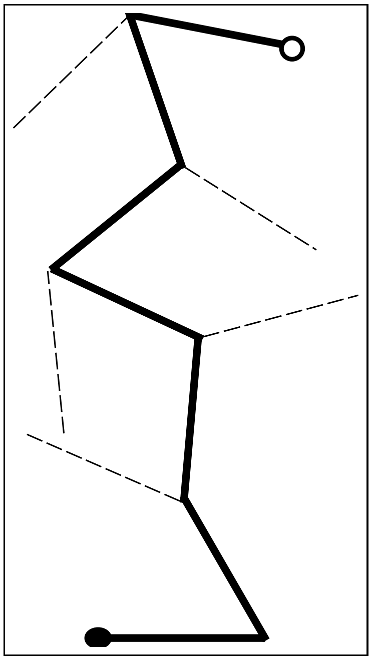

Figure 6.3: The canonical configuration of a chain. The alternative position for each edge is shown dashed.

6.2.1 Proof that a canonical chain is simple

Lemma 6.1. *In a canonical chain, no edge can point down at a slope greater than* 30° *from horizontal, and two successive edges cannot both point down.*

Proof. Suppose that we have a chain that has been placed into canonical form up to a vertex v_{i+1} with both properties true. Then edge v_iv_{i+1} must be within region R_1 as shown in Figure 6.4 (up to reflective symmetry about a vertical line). In the same figure, R_2 is the region that edge $v_{i+1}v_{i+2}$ might be found if v_iv_{i+1} happened to be vertical. One can easily see that regardless of the position of v_iv_{i+1}, $v_iv_{i+1}v_{i+2}$ must form a left turn (due to the greedy rule) and $v_{i+1}v_{i+2}$ can never point down at a slope greater than 30°. Since by definition the first edge points up, our first claim is true for all edges above it. By rotating R_2 clockwise to match any possible position of v_iv_{i+1} that points *down*, we also see that $v_{i+1}v_{i+2}$ cannot point down. □

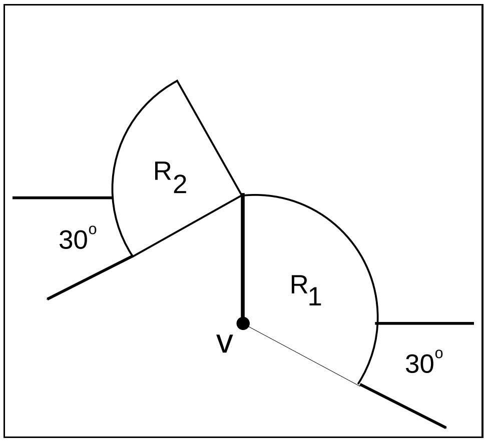

Figure 6.4: An edge following vertex v must be in region R_1. If the edge is vertical, its successor must be in region R_2.

Let the notation $v_a > v_b$ denote that v_a is higher than v_b in the vertical plane.

Lemma 6.2. *Suppose that v_iv_{i+1} points up. Then v_{i+2} is at least half a unit higher than v_i.*

Proof. Suppose without loss of generality that v_iv_{i+1} points to the right. Then $v_iv_{i+1}v_{i+2}$ form a left turn. Now we have two cases: The first case is if $v_{i+1}v_{i+2}$ also points up. Since the turn is more than 30°, the height difference between v_{i+2} and v_{i+1} is greater than $\frac{1}{2}$ $(= sin 30°)$, so the claim is true since $v_{i+1} > v_i$. The second case is if $v_{i+1}v_{i+2}$ points down. Then it is in region R_3 as shown in Figure 6.5. v_iv_{i+1} must be in region R_4. The smallest height increase occurs when the angle between the two edges is minimized. If v_iv_{i+1} is vertical, the increase is $\frac{1}{2}$. We can hold v_i fixed and rotate the two edges clockwise to observe that the height difference can only increase as v_iv_{i+1} becomes less vertical. □

Lemma 6.3. *Once a particular height h is reached by the canonical chain, the remaining chain cannot reach more than half a unit below h.*

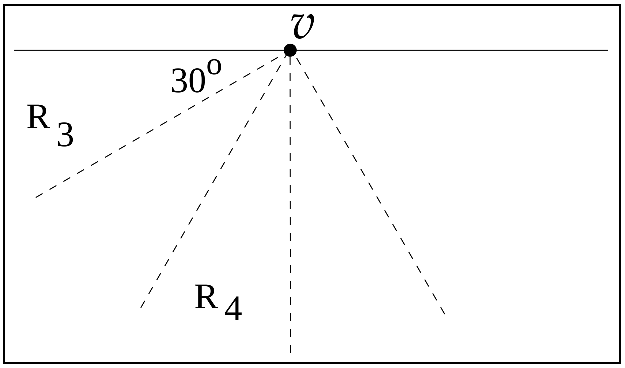

Figure 6.5: Regions of space relative to a vertex v.

Proof. By Lemma 6.1, no two successive edges can point down. Thus by Lemma 6.2, every time an edge points down, it can be combined with its predecessor for a net height increase. This proves the claim. □

The argument given in the first case of Lemma 6.2 taken together with Lemma 6.3 imply the following:

Corollary 6.4. *If e_1 and e_2 are consecutive edges that point up, no successor of e_2 can intersect e_1.*

Lemma 6.5. *A canonical configuration has the property that every third vertex has monotonically increasing height. In addition, if edge e_i points up, then v_{i+3} is at least $\frac{1}{2}$ higher than v_i.*

Proof. Consider any three consecutive edges, e_1, e_2, e_3. We will show that v_4 must always be higher than v_0. If none of the three edges point down, then the claim holds trivially.

Suppose that e_2 points up. By Lemma 6.1, a possible height decrease due to e_1 is less than $\frac{1}{2}$. By Lemma 6.2, e_2 and e_3 combine to a height increase of at least $\frac{1}{2}$.

If e_2 points down, Lemma 6.1 tells us that the other two edges point up. By Lemma 6.2, e_1 and e_2 increase height by at least $\frac{1}{2}$, and this increase cannot be negated by e_3.

If the first edge points up, then only one edge can point down so the latter can be combined with its predecessor for a net height increase of at least $\frac{1}{2}$. □

Lemma 6.6. *Every six consecutive vertices result in a height increase of at least $\frac{1}{2}$.*

Proof. Let the six edges be $e_1, \ldots, e_6$. If e_1 or e_4 point up then by Lemma 6.5 there is a triplet of consecutive edges ($e_1e_2e_3$ or $e_4e_5e_6$) that gains $\frac{1}{2}$ in height. The other triplet does not lose height, so the claim is true. If both e_1 and e_4 point down, then e_2, e_3 and e_5 point up, by Lemma 6.1. e_6 may point up or down. Thus by Lemma 6.2, pairs (e_3e_4) and (e_5e_6) each contribute a height increase of at least $\frac{1}{2}$. e_2 contributes positively, and e_1 can lose at most $\frac{1}{2}$. □

Lemma 6.7. *Edge e_{i+6} and its successors cannot intersect e_i or its preceding edges.*

Proof. The proof follows directly from Lemmata 6.3 and 6.6. □

The remaining part of this section focuses on proving that no six consecutive edges in canonical form can self-intersect. This allows us to conclude that a chain in canonical form must be simple.

From the angular restrictions of the problem definition, we have the following fact:

Fact 6.8. *No three consecutive edges can intersect.*

Lemma 6.9. *Four consecutive edges in canonical form cannot intersect.*

Proof. Let the edges be $e_1, \ldots, e_4$. By the fact above, intersection can occur only if e_1 intersects e_4. This is impossible if e_1 and e_2 both point up, by corollary 6.4.

Suppose instead that e_1 points up and e_2 points down. Without loss of generality we may assume that e_1 also points to the right, which means that e_2 points to the left and e_2e_3 form a right turn. Consider the line through e_2. e_1 is below this line, while e_3 and e_4 can only be above. Thus no intersection can occur.

The last case to consider is that in which e_1 points down and e_2 points up. Without loss of generality assume that e_1 also points to the right. It is clear that if e_2 and

e_3 form a right turn, there is no possibility of intersection. So if an intersection is to occur, e_2 must point to the right. By Lemma 6.5 $v_4 > v_1$. If e_3 points down, by Lemma 6.1 e_4 must point up, which means no intersection can occur, since $v_5 > v_4 > v_1 > v_2$. Therefore e_3 is forced to point up and e_4 is forced to point down, if there is to be an intersection.

Now, v_3 cannot be more than $\frac{1}{2}$ lower than v_1. By Lemma 6.2, v_5 must be at least $\frac{1}{2}$ above v_3 which implies $v_4 > v_5 > v_1 > v_2$, i.e. no intersection occurs. □

Lemma 6.10. *Five consecutive edges in canonical form cannot intersect.*

Proof. Let the edges be $e_1, \ldots, e_5$. By Lemma 6.9, we only need to worry about e_1 and e_5 intersecting. By corollary 6.4, there are two cases to consider:

Case 1: e_1 points up and e_2 points down. This implies that e_3 points up. Now, if e_4 points down (meaning e_5 points up), we cannot have an intersection because $v_6 > v_5 > v_2 > v_1$. Instead, if e_4 points up, we either have $v_6 > v_5 > v_2 > v_1$, or (if e_5 points down), by Lemma 6.2 v_6 has height at least $\frac{1}{2}$ more than v_4, and v_4 cannot be more than $\frac{1}{2}$ below v_2. Thus $v_5 > v_6 > v_2 > v_1$.

Case 2: e_1 points down and e_2 points up ($v_1 > v_2$).

If e_3 points down (and thus e_4 points up), we have $v_3 > v_4$, $v_5 > v_4$. By Lemma 6.5, $v_4 > v_1$, $v_5 > v_2$, $v_6 > v_3$. If e_5 points up we have $v_6 > v_5 > v_4 > v_1 > v_2$. Otherwise if e_5 points down we have $v_5 > v_6 > v_3 > v_4 > v_1 > v_2$. In either case, no intersection occurs.

On the other hand if e_3 points up, one sub-case is that e_4 points up, which means $v_4 > v_3$, and $v_5 > v_3$ by Lemma 6.2. Since $v_3 > v_0 > v_1$, no intersection occurs. The second sub-case is that e_4 points down (and thus e_5 points up). By Lemma 6.2, v_4 is higher than v_2 by at least $\frac{1}{2}$, and v_2 cannot be more than $\frac{1}{2}$ below v_0. This implies $v_5 > v_4 > v_0 > v_1$. □

Lemma 6.11. *Six consecutive edges in canonical form cannot intersect.*

Proof. Case 1: e_1 points up. By corollary 6.4 e_2 must point down if there is to be an intersection. Thus e_3 points up. If e_6 points down (implying e_5 points up) we have $v_6 > v_7 > v_5 > v_2 > v_1$ so we are done. If instead e_6 points up, we must look at e_5: if it points up, we have $v_7 > v_6 > v_5 > v_2 > v_1$, implying no intersection. Thus the only dangerous configuration remaining has e_5 pointing down (and e_4 pointing up). By Lemma 6.2, v_6 is at least $\frac{1}{2}$ higher than v_4, which is no more than $\frac{1}{2}$ lower than v_2. So $v_7 > v_6 > v_2 > v_1$.

Case 2: e_1 points down (so e_2 points up). By Lemma 6.5, $v_6 > v_3 > v_0$. Since $v_0 > v_1$, we must only prove that $v_5 > v_0$. Clearly we only worry if e_6 points up. Let us examine the pair e_4e_5. e_4 cannot point up, since then by Lemma 6.2 $v_5 > v_3$, which proves our claim. Thus e_4 points down, which means e_3 and e_5 point up. Since v_4 is at least $\frac{1}{2}$ higher than v_2, and v_2 is no more than $\frac{1}{2}$ lower than v_0, we have $v_5 > v_4 > v_0$. □

Lemmata 6.7 and 6.11 imply the following theorem:

Theorem 6.12. *A chain that is in canonical form must be simple.*

6.2.2 Lifting a chain to canonical form

Let the given chain be in the horizontal plane. We wish to use dihedral motions to reconfigure it into its canonical form in the vertical plane. We begin by lifting the first edge so that it projects vertically onto the second edge, which is still in the horizontal plane. Thus the second edge is both a part of the original configuration *and* of the canonical configuration. Now suppose that we have part of the linkage still in the original configuration, and part of it has been lifted into a vertical plane and is in canonical form. This is illustrated in Figure 6.6.

We want to move the canonical portion of the chain into a position above the next edge of the horizontal portion, as demonstrated in Figure 6.7. On the left of the

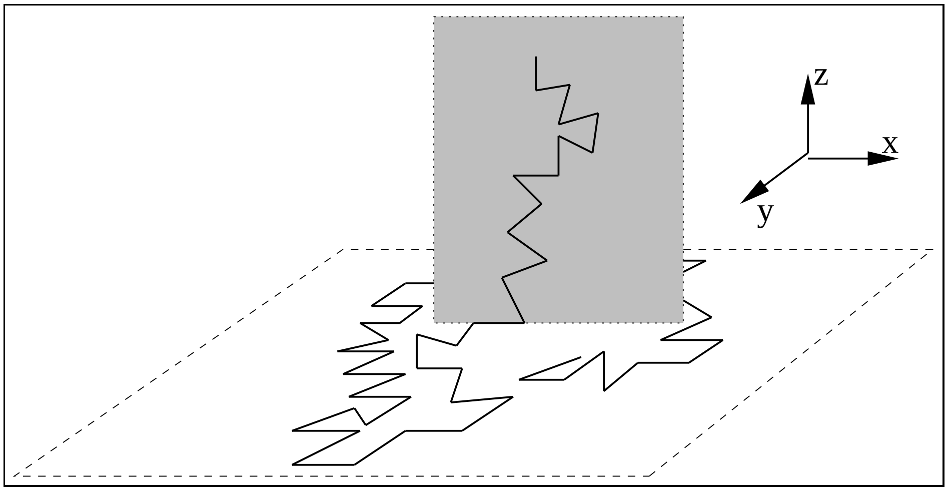

Figure 6.6: A chain that is partially in canonical form.

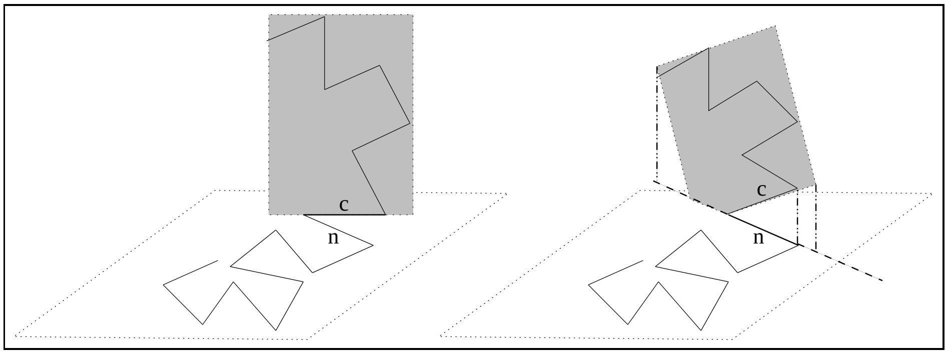

Figure 6.7: Lifting the next edge into canonical form.

figure is a partially lifted chain. The edge *common* to both planes is labelled c. This edge will be lifted so that the canonical chain will project down to the line through the *next* horizontal edge n. The result will be a configuration such as the one on the right hand side of the figure.

In Figure 6.8 we show two simultaneous dihedral motions that are performed during this operation. Edges that are already in canonical configuration remain coplanar (in a vertical plane) throughout these motions. To do this, we rotate c about n as shown on the left and at the same time we rotate the canonical plane accordingly, as

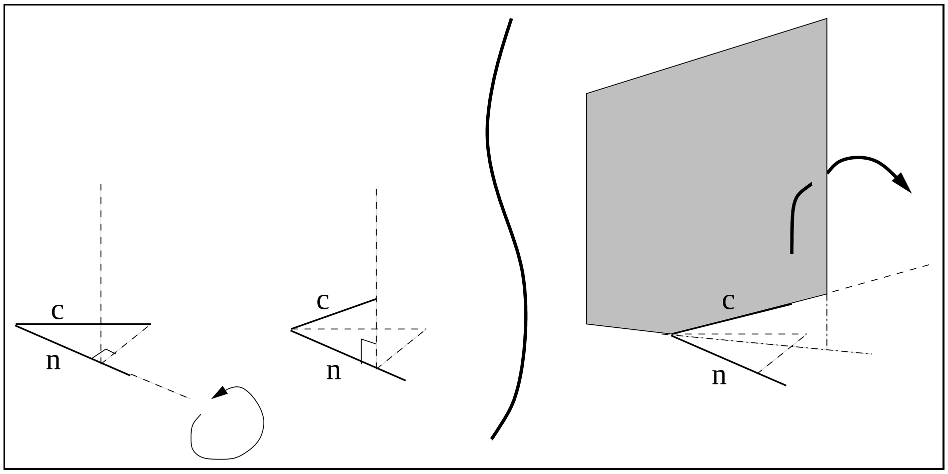

Figure 6.8: The primary dihedral motions necessary for lifting an edge.

shown on the right, so that it always projects vertically through c. We call these two dihedral motions *primary*.

While doing this, we wish to maintain the properties of the canonical chain. We only need to intervene if an edge u points directly *up* (becomes vertical) during the primary motion. At this instant the chain C_u above u may be placed arbitrarily in either of two possible positions in the canonical plane. If the overall motion is to continue, one can see that u and the edge above it will no longer satisfy the greedy property. Thus we rotate C_u about u and proceed with the primary motion, until another edge becomes vertical or c reaches its target position above n.

It is best to visualize this idea from a viewpoint that moves so that it remains perpendicular to the canonical plane. From this viewpoint, the canonical chain appears to be rotating continuously in its plane. In Figure 6.9 we show a canonical chain rotating counterclockwise in its plane. Dashed edges show the alternative position of each edge. We see that as the chain rotates, edges maintain their greedy positions until an edge u becomes vertical. If the chain continues to rotate, the property will no longer hold for the edge above u. Performing a dihedral rotation of C_u about u resolves this problem for u.

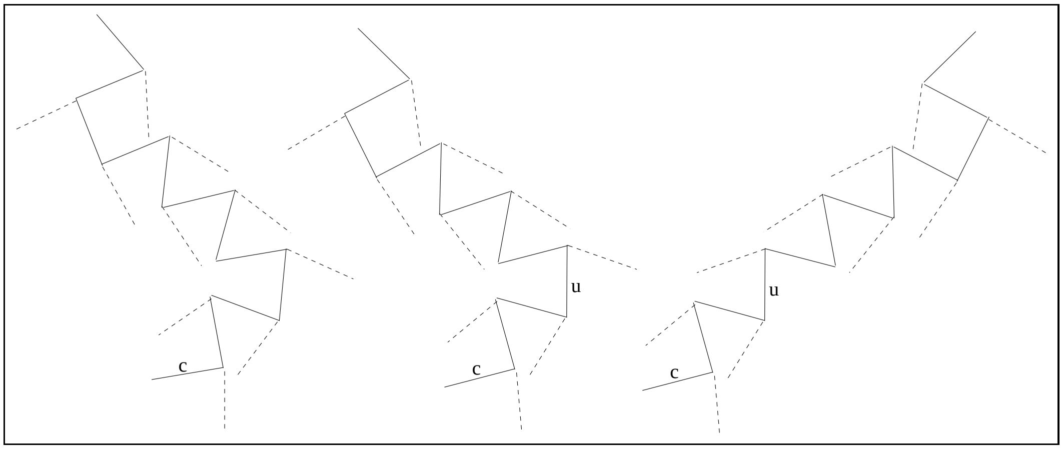

Figure 6.9: Maintaining the canonical property when lifting an edge.

We conclude with the following theorem:

Theorem 6.13. *Any two planar chains with edges of unit length and angles in the range* $(60°, 150°)$ *are flat-state connected.*

The technique described can also be used for linkages with non-acute angles and arbitrary edge lengths. The canonical configuration is defined in the same way and is clearly monotone. The primary motions of the algorithm are identical. Again, the only time that these motions must be interrupted is when an edge points vertically upward.

Theorem 6.13 should also apply to unit length chains with angles in the range $(60° + k, 150° + k)$, where k is an angle between zero and 30°. In other words the chains considered in this chapter and chains with non-acute angles are at the two extremes of the range of k.

Chapter 7

Band Unfolding

7.1 Preliminary information

After considering dihedral motions for chains, we turn to a similar method of reconfiguration for polyhedral surfaces. As described immediately below, we are not concerned with reconfiguring between two arbitrary objects here. Instead we wish to know whether all objects can be transformed to a specific state.

As mentioned in Chapter 2, it has long been an unsolved problem to determine whether every polyhedron may be cut along edges and unfolded flat to a single, non-overlapping polygon. This chapter deals with an interesting special case: That of unfolding a polyhedral *band* without overlap by cutting an appropriate single edge. Recall that a band is the surface of a convex polyhedron, enclosed between two parallel planes and not containing vertices of the polyhedron. A band and its associated polyhedron are illustrated in Figure 7.1.

This band forms the side faces of what is known as a *prismatoid* (the convex hull of two parallel convex polygons in $\mathbb{R}^3$) but the band unfolding question ignores the top and bottom faces of the prismatoid. An example was found (by E. Demaine and A. Lubiw) that shows how flattened bands can end up overlapping if the edge to be

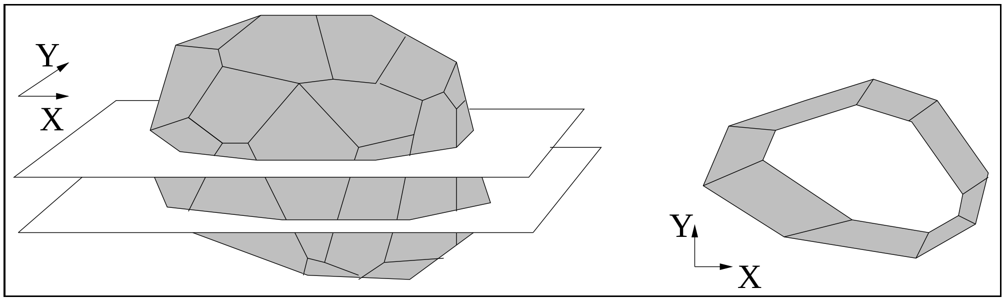

Figure 7.1: A polyhedron cut by two parallel planes, and the projection of the resulting band onto the xy plane.

cut is chosen incorrectly (see Figure 7.2).

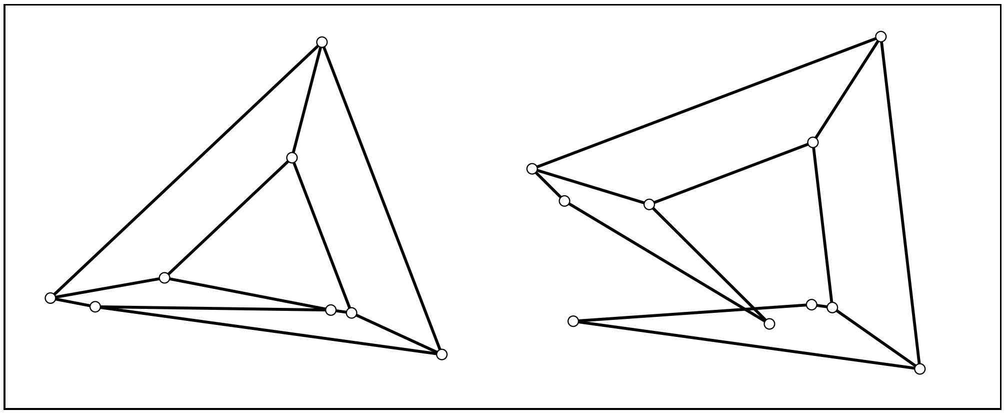

Figure 7.2: A band that self-intersects when cut along the wrong edge and unfolded. Left: projection of band. Right: self-intersecting unfolding.

There is one unfolding result that is particularly relevant to this problem, which may be interpreted as unfolding infinitely thin bands. This result states that a “slice curve”, the intersection of a plane with a convex polyhedron, develops (unfolds) in the plane without overlap (see [O’R03, O’R01]). To develop a slice curve, a cut point must be chosen, and all polyhedral faces that the curve passes through must be flattened (best visualized as a rolling motion of the polyhedron). The trace of the slice curve on the plane will not self-intersect, though its shape depends on the cut point. This result also applies to any “convex” curve on the surface of a polyhedron (in the sense that it makes only left -or only right- turns once it has been developed) [OS89]. In fact we will initially consider slice curves with this property. These proofs depend on the generalization of Cauchy's arm lemma mentioned in Chapter 2.

Thus, both the top and the bottom boundary of any band (and in fact any slice curve in between), will not self-intersect after a band has been flattened. Overlap can occur only from interaction with the cut edge, as in Figure 7.2.

Here we will prove that all bands can be unfolded by explicitly identifying an edge to be cut. We will begin with a special case of bands, defined as follows: A band is *nested* if projecting the top polygonal rim A orthogonally onto the plane of the bottom polygonal rim B results in a polygon nested inside B. For example, the band in Figure 7.1 is nested. Intuitively, we might expect to obtain a nested band if both parallel planes cut the polyhedron near its "top". We prove that all nested bands can be unfolded. Our proof provides more than non-overlap in the final planar state: it ensures non-intersection throughout a continuous unfolding motion. These results are then extended to arbitrary bands.

7.2 Properties of bands

We first define bands more formally and analyze their combinatorial and geometric structure, without regard to unfolding.

Consider a convex polyhedron P, and let $z_0, z_1, \ldots, z_m$ denote the sorted z coordinates of the vertices of P. Pick two z coordinates z_A and z_B that fall strictly between two consecutive vertices z_i and z_{i+1}, and suppose z_A is above z_B: $z_i < z_B < z_A < z_{i+1}$. The *band* determined by P, z_A, and z_B is the intersection of P's surface with the horizontal slab of points whose z coordinates satisfy $z_B \leq z \leq z_A$.

The band is a polyhedral surface with two components of boundary, called A and B. Specifically, A is the *top polygonal rim* of the band, i.e., the intersection of P's surface with the plane $z = z_A$, and B is the *bottom polygonal rim*, corresponding to the plane $z = z_B$. Both rims A and B are convex polygons in their respective planes, being slice curves of a convex polyhedral surface P. All vertices of the band are vertices of either A or B.

Every vertex of the band is incident to exactly three edges. two along the rim A or B containing the vertex, and the third connecting to the other rim. This third edge, called a *hinge*, is part of an edge of the original polyhedron P connecting a vertex of P with z coordinate less than z_B to a vertex of P with z coordinate greater than z_A. The hinge from each vertex of the band defines a perfect matching between vertices of the top rim A and vertices of the bottom rim B. This matching is consistent with the cyclic orders of A and B in the sense that, if vertex a_i of A is paired with vertex b_i of B, then the vertex a_{i+1} clockwise around A from a_i is paired with the vertex b_{i+1} clockwise around B from b_i. This correspondence defines a consistent clockwise labeling of the vertices $a_0, a_1, \ldots, a_{n-1}$ of A and the vertices $b_0, b_1, \ldots, b_{n-1}$ of B, unique up to a common cyclic shift [1] .

Each face of the band is a quadrilateral spanned by two adjacent vertices a_i and a_{i+1} on the top rim A and their corresponding vertices b_i and b_{i+1} on the bottom rim B. This facial structure follows from the edge structure of the band. Each face corresponds to a portion of a face of the original polyhedron P (so in particular it is planar). Because edges $a_i a_{i+1}$ and $b_i b_{i+1}$ lie in a common plane as well as in parallel horizontal planes, the edges themselves must be parallel. Thus every face of the band is in fact a trapezoid, with parallel top and bottom edges.

7.3 Nested bands

Next we analyze the geometric structure of nested bands in particular, still without regard to unfolding.

A band is *nested* if the orthogonal projection of A into the xy plane is strictly

[1] Throughout this chapter, indices are taken modulo n.

contained inside the orthogonal projection of B into the xy plane. (Of course, a band is just as nested if instead B's projection is contained inside A's projection, but in that case we just reflect the band through the xy plane.)

Nested bands have a particularly simple structure when projected into the xy plane. As with all bands, each face projects to a trapezoid. The unique property of a nested band is that none of its edges cross in the projection. This property follows because the projected edges are a subset of a triangulation of the projections of A and B, which themselves do not intersect by the nested property. (In non-nested bands, edges of A intersect edges of B in the projection.) Thus the projected trapezoidal faces of the band form a planar decomposition of the region of the xy plane interior to the projection of B and exterior to the projection of A.

In the xy projection, the *normal cone* of a vertex a_i of the rim A (or more generally any convex polygon) is the region between the two exterior rays that start at a_i and are perpendicular to the incident edges $a_{i-1}a_i$ and a_ia_{i+1} respectively. See Figure 7.3.

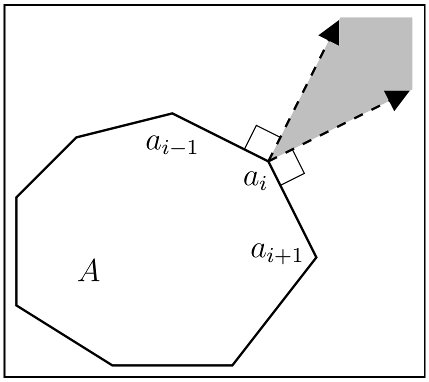

Figure 7.3: The normal cone of a vertex a_i.

Lemma 7.1. *In the projection of a nested band, not all hinges a_ib_i can be to the left (or all to the right) of their respective normal cones.*

Proof. The following proof refers exclusively to the xy projection. Suppose without loss of generality that all hinges are clockwise of their respective normal cones on the inner polygon A. For each i, define T_i to be the trapezoid with vertices $a_{i-1}, a_i, b_{i-1}, b_i$, and let h_i denote its width, i.e., the distance between the opposite parallel edges $a_{i-1}a_i$ and $b_{i-1}b_i$. See Figure 7.4. Without loss of generality, let a_ia_{i+1} be horizontal in the plane. Hinge a_ib_i is right of the normal cone at a_i. By definition of the problem, $a_{i+1}a_ia_{i-1}$ and $b_{i+1}b_ib_{i-1}$ are each left turns, so by simple trigonometry the width h_i of T_i is less than the width h_{i+1} of the clockwise next trapezoid T_{i+1}. Applying this argument to every T_i, we obtain a cycle of strict inequalities $h_0 < h_1 < \cdots < h_{n-1} < h_0$, which is a contradiction. □

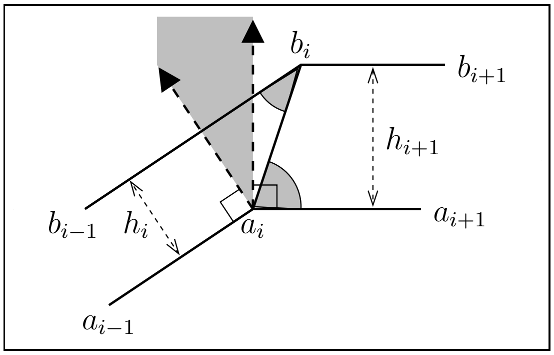

Figure 7.4: If the hinge a_ib_i is right of the normal cone at a_i, then the top shaded angle is less than the bottom shaded angle, so $h_i < h_{i+1}$.

7.4 Unfolding nested bands

After cutting a single hinge, a *flattening motion* is a continuous motion during which each face moves rigidly but remains connected to each adjacent face via their common hinge, and the final configuration is planar. If no intersection occurs during the motion, then this motion is a *continuous unfolding*. If the resulting configuration is non-self-intersecting, but intersection occurs during the motion, then we call the motion an *instantaneous unfolding* and the resulting configuration an *unfolded state*. Thus in Figure 7.2 we would say that the band has been flattened, but because it self-intersects it has not unfolded.

We now describe the flattening motion that will lead to our unfolding. The motion is based on *squeezing* together the two parallel planes $z = z_A$ and $z = z_B$ that contain the rims A and B, keeping the planes parallel and keeping each rim chain on its respective plane. At time t, the squeezing motion reduces the vertical separation between the two parallel planes down to $(1-t)(z_A - z_B)$, that is, it linearly interpolates the separation from the original $z_A - z_B$ down to 0.

The squeezing uniquely determines the hinge dihedral angles necessary to keep the vertices of the band on their respective moving planes (assuming exactly one edge of the band has been cut). See Figure 7.5 for an example of the projected motion. For nested bands, the motion increases the interior angle at every vertex of each chain in the projection. This property can be seen by examining any two adjacent faces that are being "squeezed". Both faces rotate continuously to become more horizontal. If we force one of the faces to keep its vertices in the parallel planes, but allow the second face to only follow this motion rigidly (i.e., the dihedral angle at the hinge remains fixed), then the edges of the second face no longer remain on the horizontal planes. To compensate, the second face must perform a (dihedral) rotation about the hinge. In fact the interior angle at the hinge must increase, causing the interior angles of the rims to increase.

Furthermore, because the interior angle at a vertex of a nested band can open to an angle no greater than π, the opening chains cannot self-intersect after such a motion (a fact already known from the slice-curve result mentioned earlier). For the same reason, an opening chain will always have only right turns.

As the parallel planes squeeze together, each band face remains a trapezoid in the projection. Edges $a_i a_{i+1}$ and $b_i b_{i+1}$ remain parallel and retain their original lengths throughout. Hinge projections lengthen as the band is squeezed, which causes the trapezoid angles to change. Because b_i and b_{i+1} move orthogonally away from $a_i a_{i+1}$, acute trapezoid angles increase toward $\pi/2$ and obtuse angles decrease toward $\pi/2$.

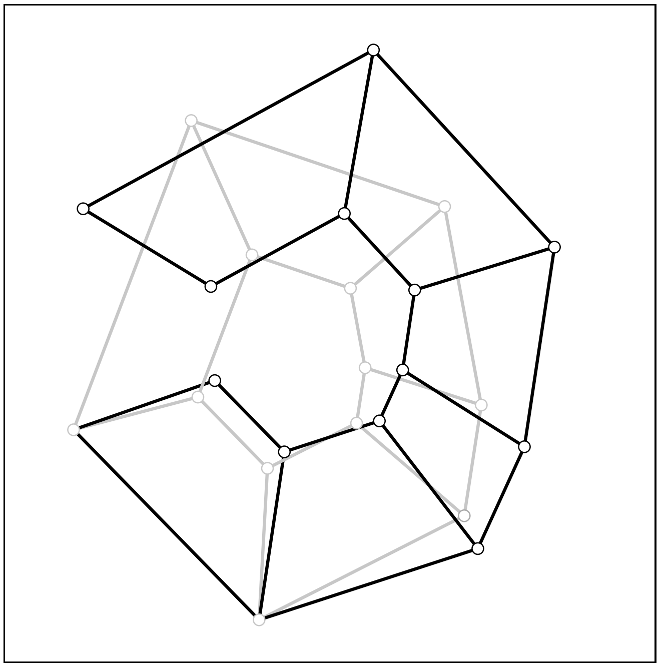

Figure 7.5: A view from above of a nested band during a squeezing motion.

The goal of this section is to show that the band does not self-intersect if we cut a specific hinge. We mention that self-intersection of the band in 3D implies self-intersection in the projection, so it suffices to prove that there is no self-intersection in the projection to establish that there is no self-intersection in 3D. It turns out that the only cause of self-intersection is the cut edge, to which we now turn our attention.

Suppose that we cut hinge $a_i b_i$ and hold $a_{i-1}a_i$ fixed along the x-axis in the positive direction. The motion separates two copies of a_i; we call the stationary one a_i, and call the moving one a^*. Correspondingly, for the outer polygon, the direction of $b_{i-1}b_i$ remains fixed (it moves away from $a_{i-1}a_i$ because the trapezoid enlarges in the projection, but remains parallel) and b^* is a "moving" endpoint. Thus the cut hinge is split into edges $a_i b_i$ and $a^* b^*$. See Figure 7.6.

We now introduce some basic terminology and properties of an opening chain in the projection. Given a chain with all right turns, the *interior angle* α_i at a vertex

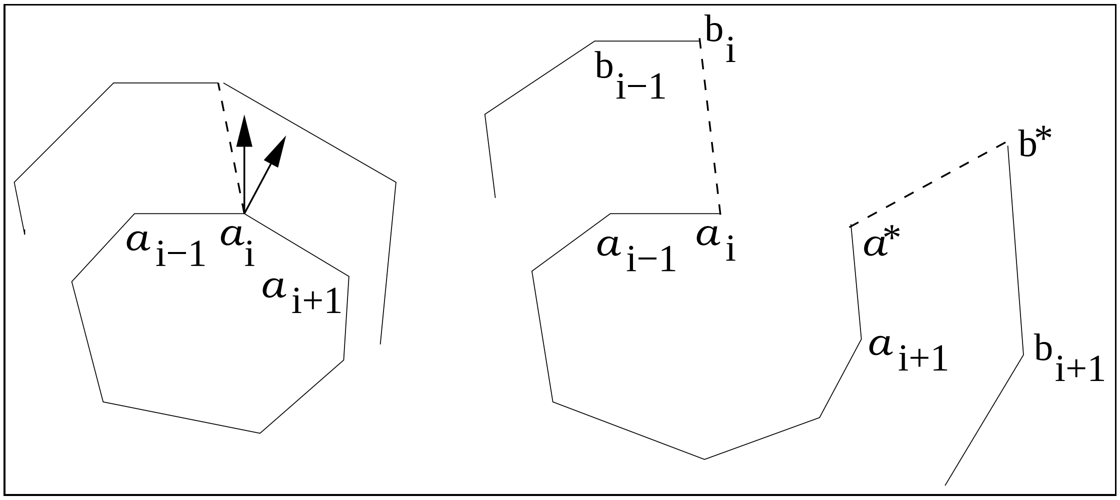

Figure 7.6: Left: projection of the inner convex chain and part of the outer chain. Hinge a_ib_i and the cone of vertex a_i are shown. Right: the result of cutting at a_ib_i and flattening.

a_i is the angle $a_{i-1}a_ia_{i+1}$ located on the right side of a_i. Let $\tau_i = \pi - \alpha_i$ be the *turn angle* at a_i. Let θ_j be the counterclockwise angle of the vector $a_j - a_{j-1}$ from the positive x-axis. If $a_i - a_{i-1}$ is fixed along the positive x-axis, then for a chain with all right turns we have: $\theta_i = 0$, $\theta_{i-1} = \tau_{i-1}$, and in general,

$$\theta_{i-k} = \sum_{j=i-k}^{i-1} \tau_j. \tag{7.1}$$

We define three classes of shapes that a chain with only right turns may have: convex, weakly convex, and spiral. Refer to Figure 7.7. A chain is *convex* if joining the endpoints with a closing segment yields a convex polygon. A chain is *weakly convex* if joining the endpoints with a closing segment yields a simple polygon with no exterior angles smaller than $\pi/2$. If a chain is not convex or weakly convex, it is a *spiral*.

Our results below apply to the projection of a band after it has been partially squeezed, and also to a band that has been flattened.

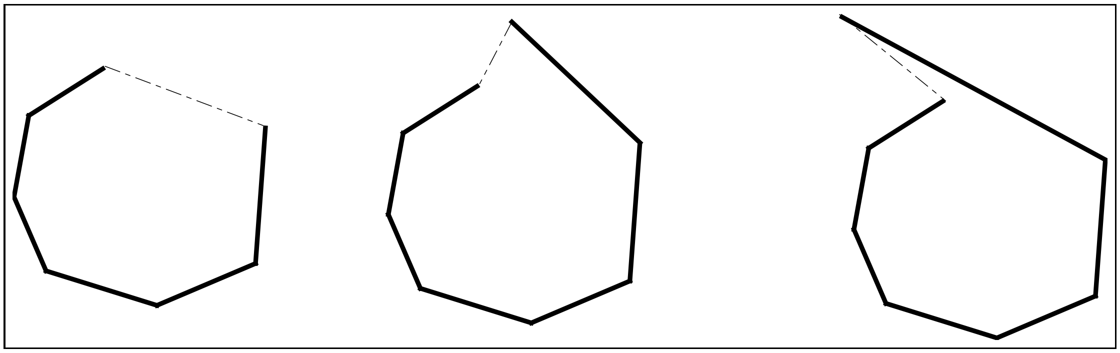

Figure 7.7: Types of chains, from left to right: convex, weakly convex, spiral. Endpoints are joined by dashed line segments.

Lemma 7.2. *Flattening a band cannot produce an inner chain that is a spiral.*

Proof. Recall that the inner chain is labeled A. Consider the line that passes through a_i and is orthogonal to $a_i a_{i+1}$ prior to any motion. Let R be the halfplane defined by this line, and which does not contain the normal cone at a_i. Let Q be the halfplane to the right of $a_{i-1}a_i$. See Figure 7.8(a).

We establish two claims about the effect of any opening motion:

1. Every edge $a_k a_{k+1}$ of A turns clockwise in the sense that the vector $a_{k+1} - a_k$ turns clockwise. In particular, $a^* a_{i+1}$ turns clockwise around a^*.

2. The vertices of A always lie in $R \cup Q$.

By the second claim, a^* cannot reach a position where it will be on the hull of a spiral, as shown on the right of Figure 7.7. By symmetry, this is enough to prove that no spiral can be created.

The motion of a^* is determined by the opening of the internal angles of all vertices on the counterclockwise path from a_{i-1} to a_{i+1}. Specifically, the opening of an angle α_j at a vertex a_j causes a^* to rotate instantaneously about this vertex a_j. Because all internal angles α_j increase, all respective turn angles τ_j decrease. Equation 7.1 shows that, if all turn angles τ_j decrease, then θ_{i+1} decreases, which corresponds to a

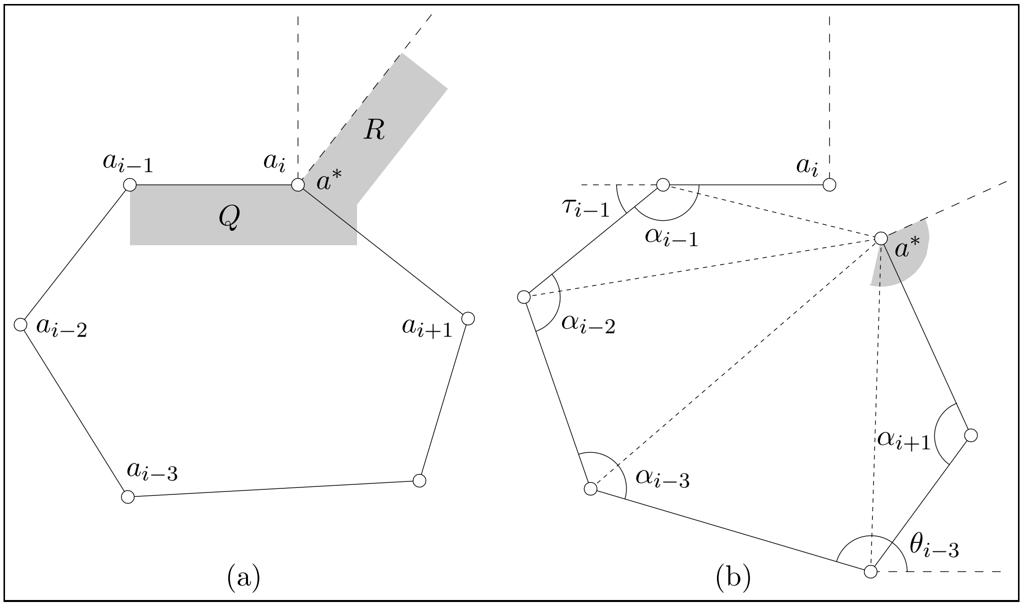

Figure 7.8: (a) Cut at $a_i = a^*$ ensures that a^* moves into $R \cup Q$; (b) After partial opening motion. The shaded cone at a^* shows the range of instantaneous vector displacements caused by rotations at all vertices.

clockwise turn of a^*a_{i+1}. This establishes the first claim, for the same holds true for all θ_k and thus all edges a_ka_{k+1}.

Turning to the second claim, we consider that each rotation centered at a vertex a_k creates a vector displacement of a^* perpendicular to a_ka^*. Every such vector "aims inside" $R \cup Q$, in the sense that any point along the extension of the vector is in $R \cup Q$. More importantly, no vector can be extended to cross through the cone at a_i. Once a^* moves, the clockwise turn of a^*a_{i+1} guarantees that again no vector aims through the cone. More specifically, a vector can either aim inside $R \cup Q$ as shown in Figure 7.8(b), or, the vector's extension can intersect a_ia_{i-1} but only if a_k is above and to the right of a^*. In any case, a^* can never enter the cone at a_i. Thus, a^* remains inside $R \cup Q$ throughout any opening motion. The same holds true for all vertices of A. □

By Lemma 7.2, we can now assume that the inner chain of a flattened nested band is either convex or weakly convex.

Lemma 7.3. *If a flattened band overlaps, then some edge crosses over one of the two cut hinge copies.*

Proof. The result mentioned earlier on slice curves shows that neither the A nor the B chain can self-intersect. Chain A cannot intersect B if they are in convex position: Let ℓ be the line that passes through segment $a_i a^*$. This is a supporting line of chain A. Now we can sweep a line parallel to ℓ within the halfplane containing A. At any position of the sweep line we obtain a segment that is in the interior of the band, then a segment between points on A, and finally another band segment. Thus no part of B can intersect a part of A.

Now suppose that A is weakly convex, and without loss of generality let a^* be on the hull. Thus a^* is to the left of the line ℓ_2 that extends through $a_{i-1}a_i$. A cannot cross B in the halfplane to the right of ℓ_2, by the same arguments given above. Consider the halfplane to the left of ℓ_2. We know that $b_{i-1}b_i$ is parallel to ℓ_2. The remaining edges of B are to the left of $a_i b_i$, and within the strip formed by ℓ_2 and the supporting line of $b_{i-1}b_i$. Thus there is no way for a part of A to intersect these edges without also intersecting $a_i b_i$. Therefore band overlap requires one or the other chain to cross through $a_i b_i$ or through $a^* b^*$ (cf. Figure 7.2). □

Lemma 7.4. *Let $T_1 = a_{i-1}a_i b_i b_{i-1}$ and $T_2 = a_i a_{i+1} b_{i+1} b_i$ be two adjacent trapezoids of a band. If a cut is made at their common hinge $a_i b_i$, then hinge copy $a^* b^*$ of T_2 must rotate clockwise with respect to hinge copy $a_i b_i$ of T_1.*

Proof. Hinge $a_i b_i$ rotates only because T_1 is flattened. Hinge $a^* b^*$ rotates because T_2 is flattened, but also because all interior angles of chain A open. We first examine the two rotations caused by the flattening of T_1 and T_2. If we were to flatten these two trapezoids without actually cutting their common hinge, then the two hinge copies, $a^* b^*$ and $a_i b_i$ (still glued together), would both undergo an identical rotation. This would also cause a^*a_{i+1} to rotate counterclockwise. Instead, by imposing that a^*a_{i+1} maintains its orientation, and that a^* remains incident to a_i, we see that a^*b^* must rotate clockwise with respect to a_ib_i. Now we may apply the third source of rotation to a^*b^*, which must also be clockwise because a^*a_{i+1} rotates clockwise, by the first claim of Lemma 7.2. □

Lemma 7.5. *If a flattening motion produces an inner chain A that is convex, then the flattened band does not overlap, i.e., it has an unfolded state.*

Proof. Suppose that we cut at a_ib_i. If the inner chain is convex, then all of chain A has moved to the right of $a_{i-1}a_i$ which means that chain A cannot intersect a_ib_i. By Lemma 7.4, a^*b^* has undergone an additional clockwise rotation with respect to a_ib_i. This relative difference in orientation and the convexity of the opened chain imply that the two hinges cannot intersect, and that b^* (and in fact, all sections of the B chain) cannot cross over a_ib_i. By symmetric arguments, nothing crosses over a^*b^*. □

Lemma 7.6. *If we cut a hinge a_ib_i that is inside the normal cone of a_i, then the nested band can be continuously unfolded.*

Proof. Apply the squeezing motion to flatten the nested band. If the opened inner chain is convex at any moment, then there is no intersection by Lemma 7.5. Now suppose that at some instant the chain is weakly convex, and without loss of generality a^* is on the hull. Let ℓ be a fixed line through a_i and the original position of b_i. Let the direction of ℓ be from a_i to b_i. Because a_ib_i can only rotate so that it becomes more orthogonal to $a_{i-1}a_i$, it will always remain within the normal cone, and more specifically, to the left of ℓ. From claim (2) of Lemma 7.2, a^* never enters the interior of the normal cone, but rather moves immediately into $R \cup Q$, where it remains throughout the motion. Hinge copy a^*b^* must rotate clockwise, because it becomes more orthogonal to a^*a_{i+1} and because all interior angles open. Thus given the relative positions of a_i and a^*, the two hinges cannot intersect.

From the arguments above, hinge copy a^*b^* cannot cross over ℓ in the region above $a_{i-1}a_i$. Since a_ib_i was in the normal cone of a_i, angle $a_{i+1}a^*b^*$ is obtuse. Thus a^*b^* cannot rotate enough clockwise to intersect ℓ in the region below $a_{i-1}a_i$.

The intersection of ℓ with the band produces one line segment, $a_\ell b_\ell$ (other than the point a_i). The line ℓ divides the band into two disjoint bands whose interior chain is convex, and so by the arguments of Lemma 7.3, they do not self-intersect.

By symmetric arguments there can be no intersection if the partially opened chain is weakly convex with a_i on the hull. □

Note that Lemma 7.6 implies that any band with an acute interior angle can be unfolded, for there must be a hinge inside the normal cone at an acute angle: see Figure 7.9. Similarly, a geodesic cut (not necessarily along a hinge) perpendicular to supporting lines to both the inner and outer chains leads to a continuous unfolding.

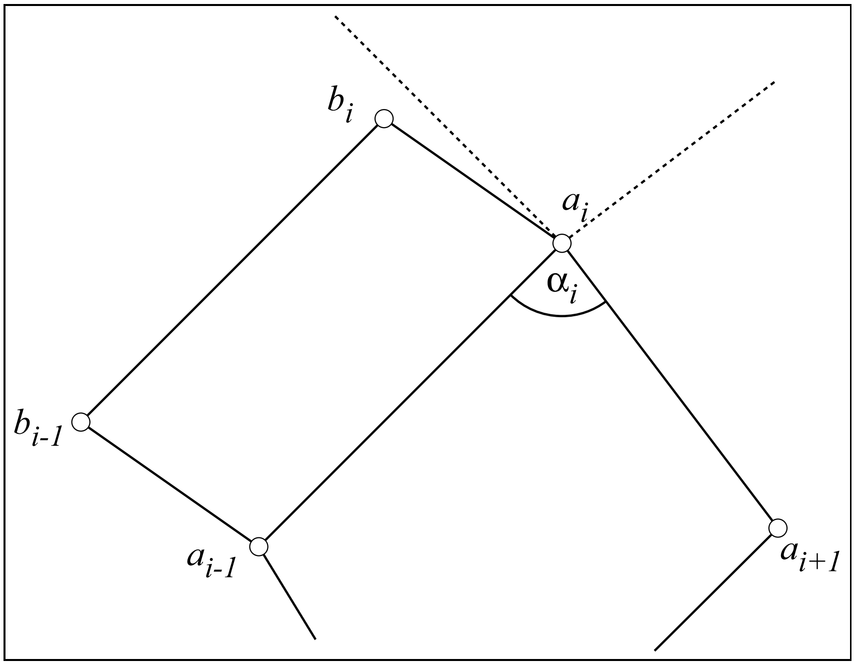

Figure 7.9: $\alpha_i < \pi/2$. If the hinge a_ib_i is outside the normal cone, it is impossible to complete a trapezoid whose base is parallel to a_ia_{i+1}.

We now characterize the types of chains that may be obtained after cutting at a_ib_i and flattening (or partially flattening by applying a squeezing motion). We say that a chain is "safe" if it could not be the inner chain of a self-intersecting band. Thus, by Lemma 7.5, all convex chains are safe. Weakly convex chains are not necessarily safe. Let us subdivide this class of chains into *L-weakly convex* and *R-weakly convex*, depending on which endpoint is on the hull (clearly exactly one of the two endpoints must be on the hull). If a_i is on the hull then the chain is L-weakly convex. Otherwise, if a^* is on the hull, the chain is R-weakly convex. A chain may open to a weakly convex position and be safe, as seen in Lemma 7.6. In fact, for an R-weakly convex chain, an intersection cannot occur if a_ib_i was initially to the left of the normal cone at a_i. In this case, a^* must be to the right of a_ib_i. Since a^*b^* rotates more clockwise than a_ib_i, no intersection can occur (we may form a line through the final position of a_ib_i and repeat the arguments of Lemma 7.6. So we say that a chain is "unsafe" if it is R-weakly convex and in the initial projection a_ib_i was to the right of the normal cone at a_i (see Figure 7.10). By symmetry a chain is unsafe if it is L-weakly convex and in the initial projection a_ib_i was to the left of the normal cone. We note that even under these conditions there may be no intersection at any time during a flattening motion. In other words, the term "unsafe" serves just as a warning.

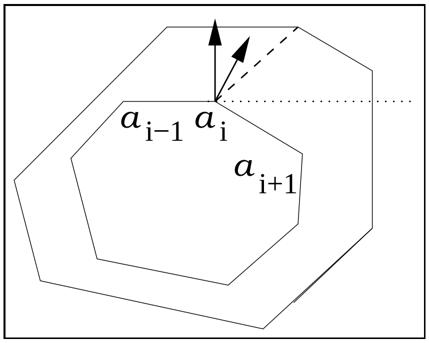

Figure 7.10: After cutting at a_i, the inner chain will become R-weakly convex if a^* ends up above the dotted line. In this case the cut is labeled "unsafe" if hinge a_ib_i (shown dashed) is to the right of its normal cone.

Clearly if a band has no unfolded state then all vertices are associated with unsafe openings. By Lemma 7.1, somewhere there is a vertex a_k whose hinge is counterclockwise of the normal cone at a_k, while the hinge at a_{k+1} is clockwise of its respective cone. For the cuts at both vertices to produce unsafe inner chains, cutting at a_k must produce an L-weakly convex chain, while cutting at a_{k+1} must produce an R-weakly convex chain (see Figure 7.11).

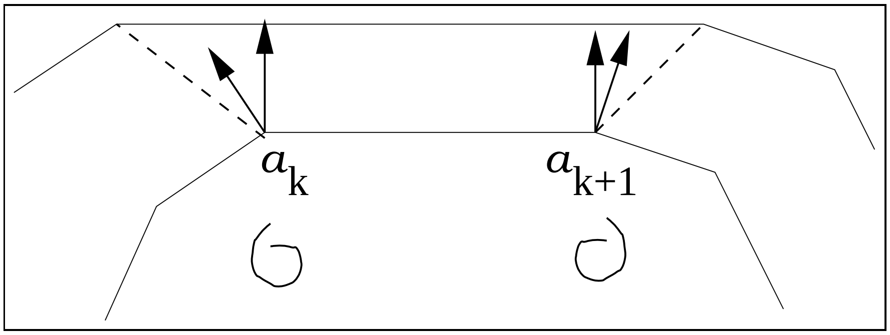

Figure 7.11: Two successive vertices, a_k and a_{k+1}, whose cuts produce different weakly convex chains (indicated by the curves below the vertices).

Lemma 7.7. *At least one of the cuts at a_k or a_{k+1} (defined above) must result in an opened chain that is safe.*

Proof. Let us begin by cutting at a_{k+1} and flattening. Hold $a_k a_{k+1}$ fixed and open all angles. Assume that the opened chain is unsafe. This means that newly created a^* must end up in the upper-right quadrant of a_{k+1}. Now we make a new cut at a_k, and translate the entire opened chain (except the fixed edge) so that a^* re-attaches to a_{k+1}. We let the translated copy of a_k retain its label, and call the fixed edge $a^* a_{k+1}$. Notice that a_k must be in the lower left quadrant of a^* (see Figure 7.12).

Now we have a new opened chain, except that we have not taken care of the openings at the angles of a_k and a_{k+1}. Because $a_{k+1}a_{k+2}$ (previously $a^* a_{k+2}$) had rotated clockwise in the first opening, and we have merely translated it back, we must rotate it counterclockwise to return it to its initial orientation. We must then further

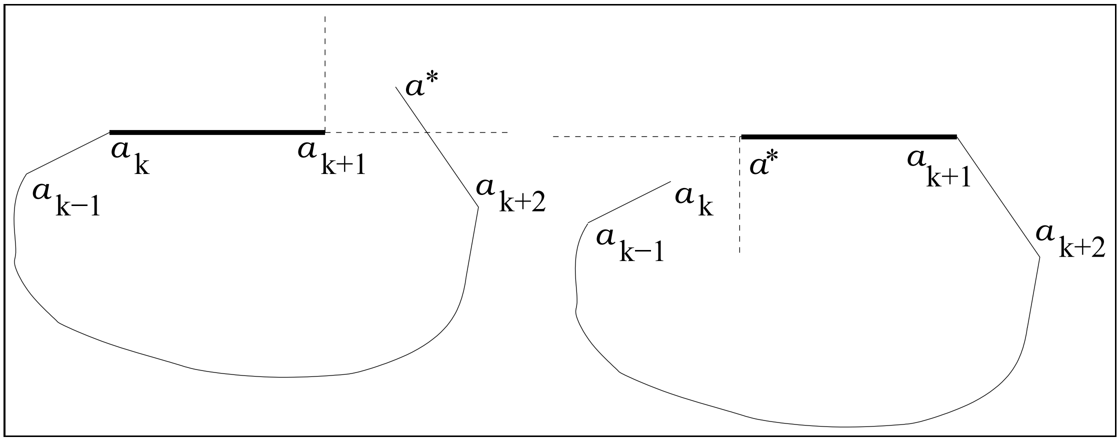

Figure 7.12: Left: an opened chain. Right: translating part of the chain so that the cut vertex is switched. This is a new opened chain, except for the angle at a_{k+1}.

rotate it counterclockwise in order to open the interior angle at a_{k+1}. The entire chain will rotate rigidly as well. Thus a_k cannot cross into the upper-left quadrant of a^*. Now notice that during the first opening, edge $a_{k-1}a_k$ rotated clockwise, due to the opening of the angle at a_k. So we might expect that in order to compensate for this in our final diagram we should rotate $a_{k-1}a_k$ counterclockwise (which might cause a_k to go above the horizontal line). After all, if a cut is made at a_k, then $a_{k-1}a_k$ must rotate counterclockwise from its initial position, but now it is clockwise. However, because the opening of the angle at a_{k-1} was included in the first opening, and this has not been tampered with, then edge $a_{k-1}a_k$ must be in its correct position. The counterclockwise motion produced by adjusting the angle at a_{k+1} is enough to make the direction of $a_{k-1}a_k$ more counterclockwise than it was initially. We conclude that cutting at a_k leads to either an R-weakly convex opening or to a convex opening. Therefore cutting at a_k cannot be unsafe. □

Because we can always find a vertex to cut so that the inner chain opens to a position that is safe, we can always find an edge to cut along so that a nested band has an unfolded state.

Theorem 7.8. *Every nested band has an unfolded state.*

Theorem 7.9. *For every nested band there exists a continuous unfolding motion.*

Proof. Consider the squeezing motion that we have defined. Parameterize any point p on the band by its original height z_p divided by the height z of the original band. After partially squeezing the band to height z_S , the new height of p will be $z_S \frac{z_p}{z}$. So all the points with the same original height have the same new height at some time of the squeezing motion.

Now consider two points p and q, that intersect at some instant during a squeezing motion. The two points are in some horizontal plane H when they intersect. Because p and q both started out in another horizontal plane H^*, and had the same height at all times throughout squeezing, they are part of a curve that satisfies the conditions of the generalized Cauchy arm lemma. Thus at any moment this curve satisfies the conditions of a developing slice curve, given in [O'R03]. We conclude that no intersection can occur until the final flattened configuration of the band, which is a singularity where the above arguments do not apply. However, by Theorem 7.8 we know that some cut exists that produces an unfolded state. Therefore by making the same cut and applying the squeezing motion, we obtain a continuous unfolding of the band. □

7.5 Unfolding general bands

Here we examine bands which generate arbitrary projections. Non-nested bands may have a more complicated appearance (as do the proofs used here to show that they have unfolded states), but in a way unfolding becomes even more likely because the cut hinge copies move further apart than in the nested case.

As mentioned in section 7.2, the projection of a non-nested band can be decomposed into trapezoids. The difference from the nested case, is that now some trapezoids will overlap, as shown in Figure 7.13(c and e). It is clear that one of two overlapping trapezoids must "flip over" when the band is flattened, as shown in parts (d) and (f) of the same figure. It is possible that two trapezoids that appear to be locally nested will both flip over, as shown in parts (a) and (b).

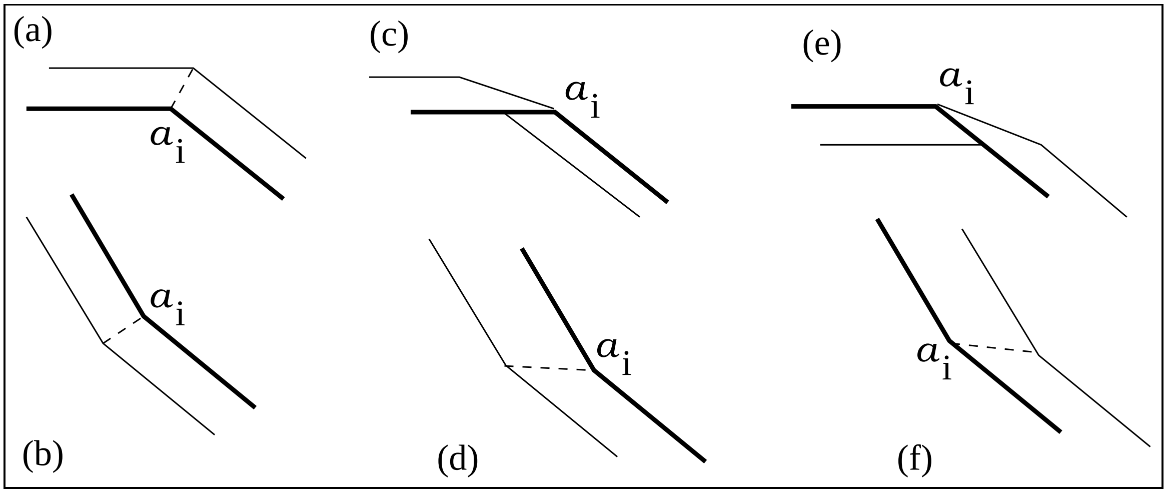

Figure 7.13: (a,c,e) : Local views of a general band. (b,d,f) : Possible results of flattening in each respective case.

As can be seen in the figure, a major difference from the nested case is that an interior angle may open to more than π. The following lemma proves that the smaller angle at any vertex will never be smaller than the original interior angle.

Lemma 7.10. *Let α_i be the internal angle at a_i prior to flattening. Then the smaller of the two angles $a_{i-1}a_ia_{i+1}$ measured after flattening must be greater than α_i. Furthermore, edge $a_{i-1}a_i$ must rotate clockwise with respect to edge a_ia_{i+1}.*

Proof. If the cone $a_{i-1}a_ia_{i+1}$ contains $b_{i-1}b_ib_{i+1}$ (or vice versa), then locally the band resembles the nested case as shown in Figure 7.13a. If the top side of these two faces remains on top in the flattened state, then the interior angle at a_i (and b_i) will open to no more than π, just as in the nested case. On the other hand, if the two faces flip over, then it is sufficient to reflect our viewpoint through the xy-plane to see that the first claim is true. In either case, by holding edge $a_i a_{i+1}$ fixed, it is easy to see that the second claim is also true.

The two overlapping cases of Figure 7.13 are symmetric, so only case (c) is discussed here. Without loss of generality, suppose that the trapezoid F_1 containing $a_i a_{i+1}$ will not flip over (if it does, we can reflect our viewpoint through the xy-plane). Perform a dihedral rotation about the hinge $a_i b_i$, so that vertices a_{i-1} and b_{i-1} reach their original z-coordinates. In other words, in the projection we just reflect one face about the hinge. The resulting configuration resembles that shown on the right side of Figure 7.14. Notice that the smaller angle at a_i is shown on both sides: on the left, this is the interior angle; on the right, in this particular case, we see that it has switched sides.

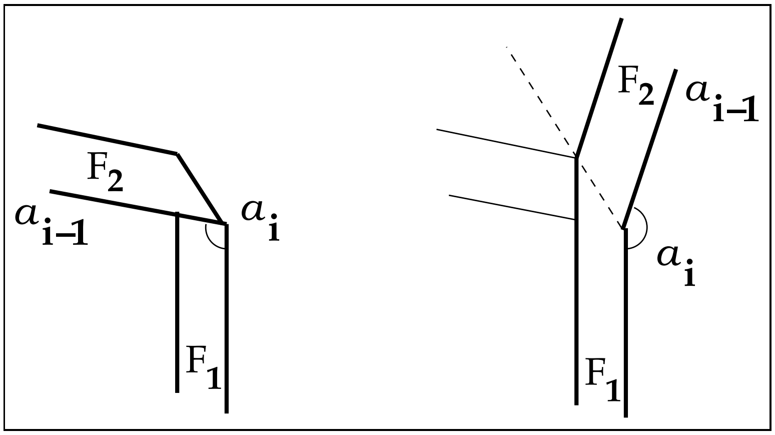

Figure 7.14: Left: two overlapping faces. Right: Performing a rotation about the hinge.

Since we are dealing with overlapping faces, the hinge must make a left turn following edge $a_{i+1}a_i$. When we flatten F_1, the left turn is preserved. This implies that the smaller angle at a_i cannot become smaller than the original interior angle, after performing the dihedral motion (reflection). This motion eliminates the overlap between the two faces. Thus we may proceed to flatten both, and rely on the preceding case mentioned above to conclude that this cannot decrease the smaller angle at a_i.

The direction of the hinge relative to $a_{i+1}a_i$ guarantees that $a_{i-1}a_i$ will rotate clockwise after the dihedral motion. Flattening can only cause further clockwise rotation if the smaller angle has not switched sides. If it has switched sides as shown in the figure, then it is clear that $a_{i+1}a_ia_{i-1}$ will still be a right turn after flattening. We conclude that the interior angle α_i cannot open by no more than $2(\pi - \alpha)$. □

Since the smallest angle at all a_i occurs prior to flattening, each convex chain cannot self-intersect. Proof of this statement is omitted, since the claim is equivalent to the slice-curve result mentioned at the beginning of this chapter. Therefore, just as in Section 7.4, we just need to focus on the hinges when studying possible overlap.

Lemma 7.11. *All edges of a polygon in the projection will rotate clockwise after flattening.*

Proof. As shown in Lemma 7.10, the angle adjustment at any vertex a_i leads to a clockwise rotation of edge a_ia_{i-1} (keeping $a_{i+1}a_i$ fixed). The composition of these rotations leads to a clockwise rotation for any edge. □

As in the nested case, we claim that if there is an acute interior angle, we can safely cut there. The logic remains the same, and it suffices to examine the relation between the hinge and the normal cone. Thus from now on we assume that there are no acute interior angles. This claim is not critical for the proofs to follow, but it makes the description simpler.

It is necessary to discuss a few more differences between nested and non-nested bands, before proving the main results.

Consider $a_{i-1}a_i$ and a_ia_{i+1}, two adjacent edges of the inner polygon in the nested case. Suppose that we cut at a_i, keeping the first edge fixed, and flatten the band. This cannot cause a_ia_{i+1} to rotate clockwise by more than 2π, since each angle opens to π at most. As we have just seen, in the general case this last statement is no longer necessarily true. Actually, there is no longer a clear definition of “inner” and “outer” polygon in the projection.

Figure 7.15 informally shows what can happen to flattened bands in the general case. At the top-left is an unflattened band (the illustration is not meant to show that it is nested). The hinge that will be cut is shown. Without loss of generality suppose that the face containing $a_{i-1}a_i$ will remain in its position without flipping over after flattening. We temporarily assign the term ”inner chain” to the one containing $a_{i-1}a_i$. Notice that $a_{i+1}a_i$ may initially point anywhere down and to the right, as indicated next to the band. At the bottom-left of the figure is a band that has been flattened, and the inner chain is in a position where overlap is possible (based solely on the orientation of the moving end, $a_{i+1}a^*$). Here, $a_{i+1}a^*$ has rotated clockwise from its original direction (that of $a_{i+1}a_i$). As long as it points to the left, as shown, overlap is possible. We allow the “inner chain” to retain its label, since apart from the non-convex shape of the band, the ends appear similar to those in the nested case. In other words, potential overlap can still be determined based on the inner chain. At the top-right of the figure, the moving end has rotated clockwise, enough so that it cannot possibly be dangerous: $a_{i+1}a^*$ no longer points to the left, so no overlap can occur. Each of these situations may arise in the nested case (ignoring the non convex shapes illustrated).

The bottom-right of Figure 7.15 shows a configuration that cannot occur in the nested case, but must be considered in the general case. Notice that the arrows have been switched to the other chain. Here, $a_{i+1}a^*$ has rotated so much clockwise that it now points to the left again. What was considered to be the “inner” chain has now been switched to become the “outer” chain. This configuration is symmetric to that at the top-right. In order for overlap to become possible again, it would be necessary for $b_{i+1}b^*$ to rotate even more clockwise, past the vertical position, in a configuration symmetric to that at the bottom-left of the figure.

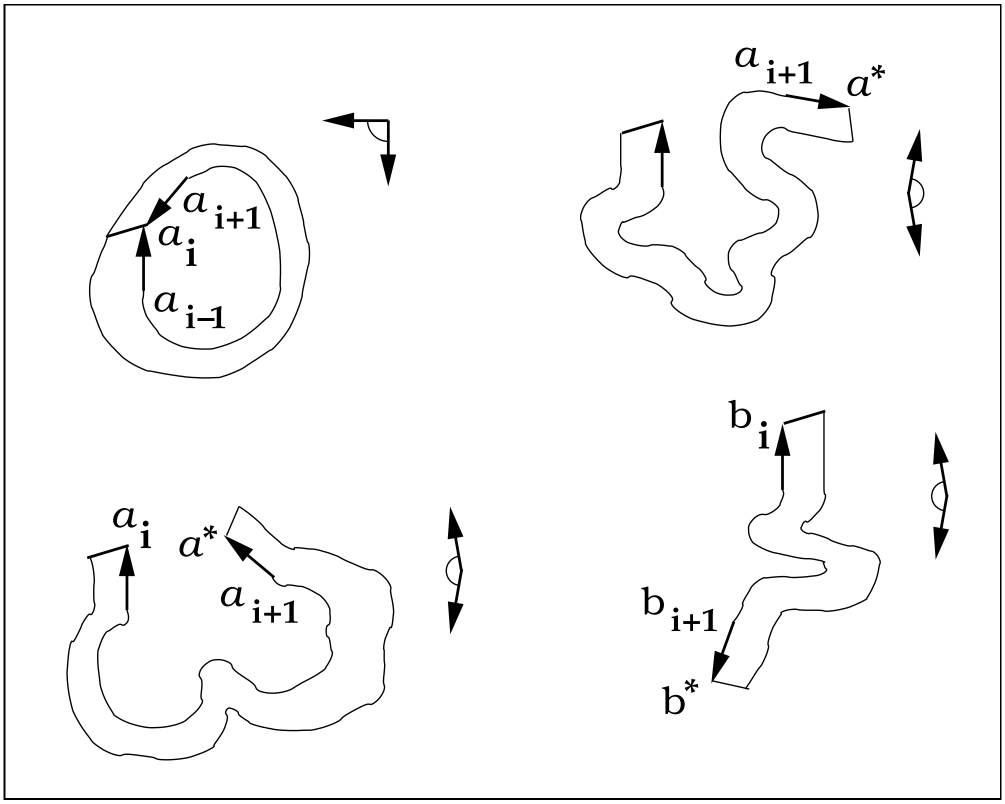

Figure 7.15: Examples of flattened bands in the general case.

We have seen that every flattening results in the moving end $a_{i+1}a^*$ rotating clockwise, but now it can theoretically perform a total turn of 4π. Let us say that if the inner chain becomes an outer chain as shown in Figure 7.15, then the band has "inverted". The minimum rotation (of the moving end) required for inversion is $\frac{3\pi}{2}$ (assuming no acute angles exist; otherwise it would be $\frac{\pi}{2}$). A rotation of at least 2π ensures inversion.

Lemma 7.12. *If no cut results in an inverted chain then a non-nested band has an unfolded state.*

Proof. Select a vertex v as shown at the top-left of Figure 7.16. This vertex belongs to two faces that overlap. As we traverse the band in a clockwise manner, face F_2 is hidden by F_1 in the neighbourhood of v. Without loss of generality assume that the side of F_2 that is visible in the original projection will remain so in the flattened state. Thus F_1 will flip over. Consider the chain that v belongs to. For a possible overlap after flattening, this chain must open as shown at the top-right of Figure 7.16. This is similar to the L-weakly convex opening for nested bands. This is the only way that the hinge might become involved in an intersection.

Now, using the same figure, we see that by following the band counterclockwise, we will encounter two faces as shown at the bottom-left. By continuity, F_3 will flatten as is, and F_4 will flip over. Vertex u is along the same chain as v. If we were to cut at u, the only way to get an overlap is if the type of opening is *opposite* from that at v, as shown on the bottom-right of the figure (this is analogous to R-weakly convex openings).

Therefore there must exist adjacent vertices, between v and u, that have different types of openings if both are to be unsafe. We may repeat the arguments of Lemma 7.7 to contradict this, just as in the nested case. □

By simply reversing the roles of each chain, the preceding Lemma implies the following:

Corollary 7.13. *If all cuts result in inverted chains then a non-nested band has an unfolded state.*

Lemma 7.14. *If some cuts lead to inverted chains and some do not, then an unfolded state exists.*

Proof. Consider two adjacent vertices, one that inverts the chain and one that doesn't. As described earlier, if a chain does not invert, the moving end can rotate by at most π and still be in a position that causes overlap. Let us begin by cutting at the vertex a_i that does not cause inversion. Then the "inner" chain is A. Let a_{i-1} be the inverting vertex, adjacent counterclockwise to a_i. When we cut at a_i, the moving end rotates

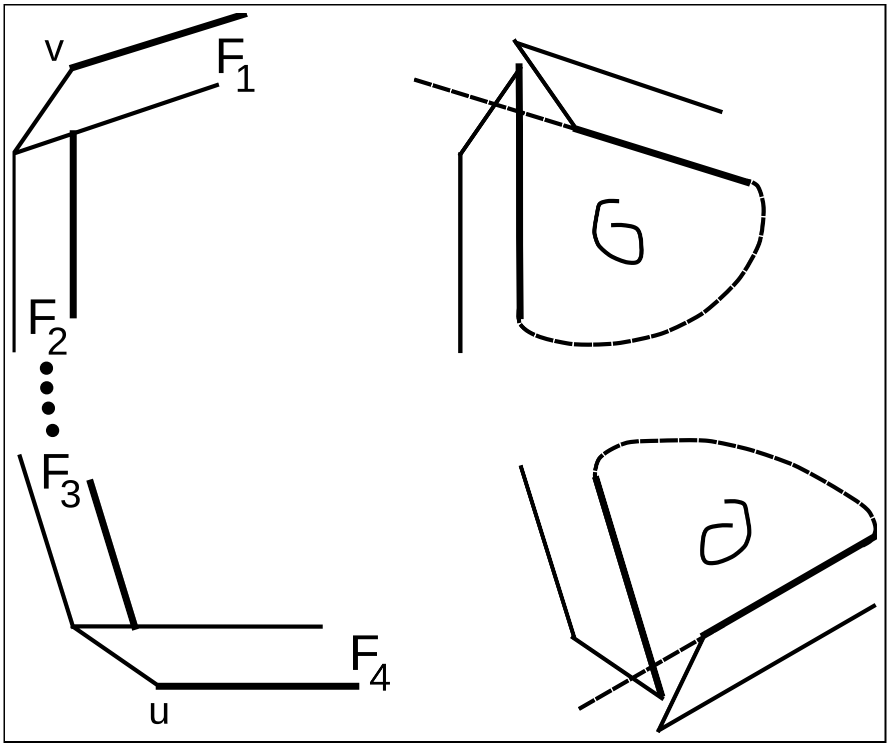

Figure 7.16: How to select a cut if no inversions occur.

by at most π, and edge $a_{i-2}a_{i-1}$ may also rotate (only clockwise) by some amount between zero and π (see Figure 7.17).

Now, as in the proof of Lemma 7.7, we unglue the hinge at a_{i-1}, and re-attach vertex a^* to its original location with a rigid motion. Then we can restore the original direction of edge a_ia_{i+1}, with a counterclockwise motion (that is less than π). This is shown in Figure 7.18.

We must also account for the angle opening at a_i, which could be at most a counterclockwise rotation of π since the angle is not acute and cannot decrease in value. We note that the re-gluing and the new angle adjustment can add up to at most a counterclockwise rotation of $\frac{3\pi}{2}$, even though each on its own could be up to π. Thus, the new end, $a_{i-2}a_{i-1}$, which could not have been pointing down initially (because we assume no acute angles exist), can now rotate counterclockwise to point

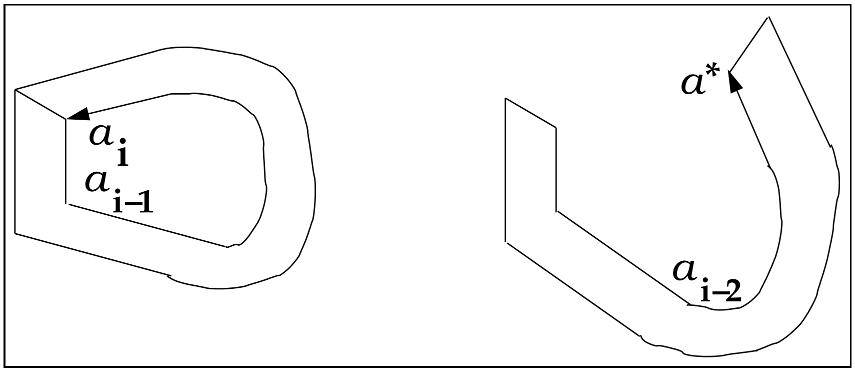

Figure 7.17: Cutting at a non-inverting vertex.

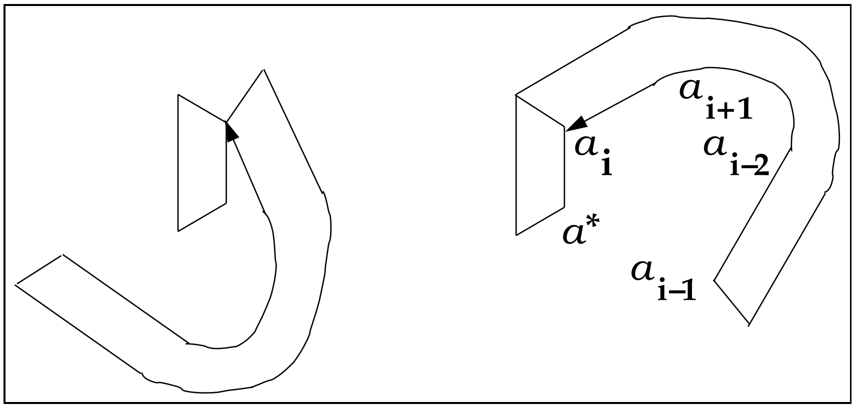

Figure 7.18: Left: ungluing hinge $a_{i-1}b_{i-1}$ and translating a^* to its original position. Right: Restoring the original orientation of a_ia_{i+1}. Notice that vertices are relabeled.

at most straight up, which is nowhere near what is necessary for it to cause overlap in an inverted configuration. In fact, given the assumption that no acute angles exist, the chain cannot even reach an inverted configuration. This contradiction implies that cutting at one of the two vertices produces a safe chain. □

Figure 7.19 shows the band after the adjustment of the angle at a_i has been made. On the right is a rotated copy, which brings things into the perspective of Figure 7.15. In this copy, we view edge $a_{i-2}a_{i-1}$ as fixed, and see that edge a_ia^* would have to rotate clockwise by even more than it already has, in order to invert.

We conclude the following:

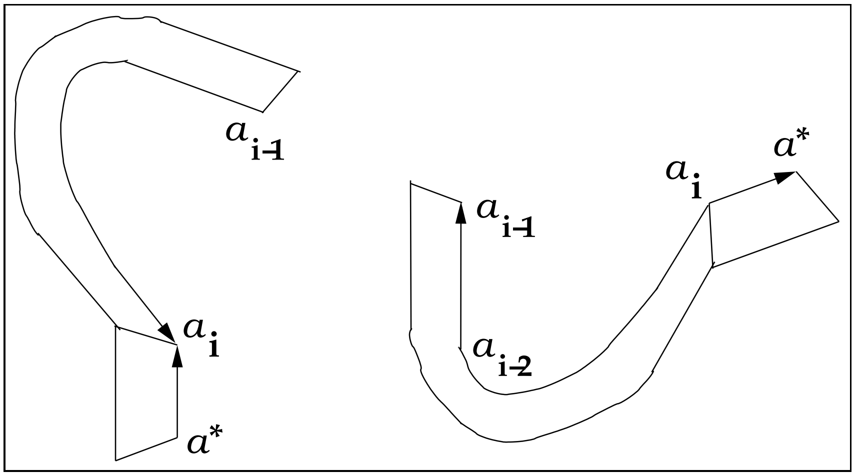

Figure 7.19: Adjusting the angle of a.

Theorem 7.15. *Every band can be cut along one of its edges to produce an unfolded state.*

It is clear that the squeezing motion of the previous section cannot be used to obtain a continuous unfolding of non-nested bands. Instead, we propose the following continuous unfolding motion which consists of $n-1$ separate moves that "peel" a band.

After cutting a hinge a_ib_i that is safe, we start by performing a dihedral rotation about hinge $a_{i-1}b_{i-1}$ so that its two adjacent trapezoids become coplanar. No intersections can happen during this move because, at any given time, trapezoid $a_{i-1}a_ib_ib_{i-1}$ is in a supporting plane of the remaining polyhedron. The next move is a dihedral rotation about hinge $a_{i-2}b_{i-2}$. In general, before we perform a dihedral rotation about $a_{i-k}b_{i-k}$, we can divide the band into two components: The first component consists of all trapezoids incident to hinges at vertices $a_{i-1}, a_{i-2}, \dots, a_{i-k+1}$, and has already been positioned into the plane of trapezoid $a_{i-k}a_{i-k+1}b_{i-k+1}b_{i-k}$. The second component consists of all remaining trapezoids which are still in their original positions.

Theorem 7.16. *The peeling motion (described above) is a continuous unfolding motion. Thus all bands can be continuously unfolded.*

Proof. Clearly no trapezoids within the second component can intersect. No intersections can occur within the first component since they would also be present after the final move when the band has been flattened. This would contradict the safe instantaneous unfolding by cutting at a_ib_i. No intersections can occur between trapezoids in separate components, since the first component is in a supporting plane of the original polyhedron. Finally, during each motion no intersections can occur, since the plane containing all trapezoids of the first component rotates about an edge of the original polyhedron and remains a supporting plane. □

7.6 Bands containing vertices

In a *closed band*, vertices are allowed on the parallel planes of the slab. The main difference of closed band projections is that vertices may attach to more than one hinge. Thus, apart from trapezoids, the decomposition of a band can include triangles (see Figure 7.20).

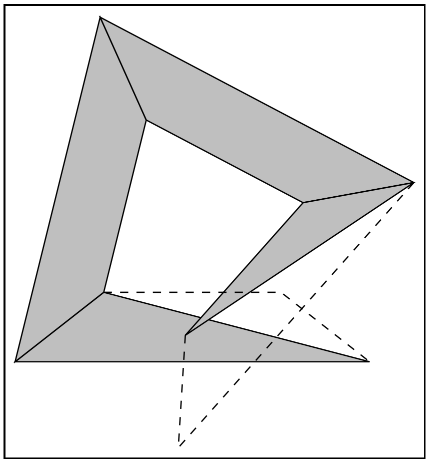

Figure 7.20: A closed band in an overlapping flat configuration.

All arguments concerning the opening of interior angles and the clockwise rotation of edges still hold. Below we will discuss nested closed bands, more for the sake of improving intuition than to provide detailed proofs. We claim that the problem does not change at all if multiple hinges are attached to vertices of the outer polygon in the projection. However, some care is required if a vertex of the inner polygon has multiple hinges.

Notice that all interior angles of the inner chain in Figure 7.20 are acute. Why then, was there overlap after selecting such an angle? The answer is that the correct vertex was selected, but not the correct hinge attached to that vertex. Had the other edge been cut at the same vertex, the band would have unfolded. This is explained below.

It is still true that a nested band with an acute angle at vertex a_i can be unfolded by cutting a hinge at a_i. Before, we relied on the fact that the (only) hinge at a_i had to be inside the normal cone. Now this is not necessarily true, if a_i has multiple hinges, as shown in Figure 7.21. This example also demonstrates that convex openings are not trivially safe. However, in such cases, regardless of the type of opening that the inner chain might make after cutting at a_i, we can choose a hinge to cut on the appropriate side of the normal cone.

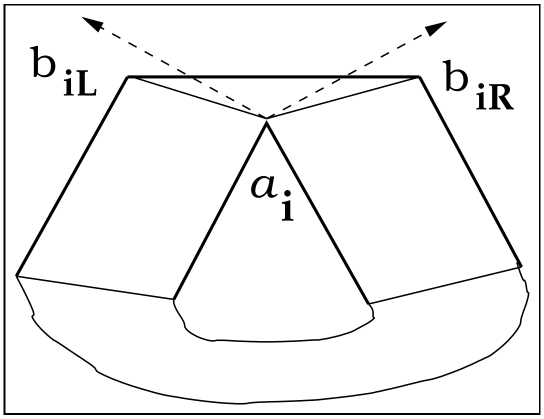

Figure 7.21: A difference between closed and open bands.

Let us focus on the first hinge to each side of the normal cone, a_ib_{iL} on the left and a_ib_{iR} on the right. Consider a line ℓ through a_i, parallel to the edge $b_{iL}b_{iR}$. The hinges are above the line, and the inner polygon is below the line. Cut the hinge a_ib_{iR} and keep the edge counterclockwise of a_i fixed in the plane. There is a danger of overlap only if the moving end crosses above ℓ. But now we can repeat the same arguments as in Lemma 7.7 to show that cutting hinge a_ib_{iL} is safe.

The arguments above work for any ℓ that separates the inner polygon and the two chosen hinges. However, our choice of ℓ was not arbitrary. Had we produced a slightly narrower band from the same polyhedron, there would have existed a new edge on the inner chain, parallel to $b_{iL}b_{iR}$. The creation of this new edge would provide the necessary conditions for Lemma 7.7. It makes sense that all previous results remain valid for closed bands , since we can approach as close as we like towards a closed band without including vertices on the two planes. In fact, as we have shown above, a good way to choose the cut vertex on a closed band is to examine what happens in the limit of its associated "open" band. For example in the case of the band given in Figure 7.20, we could lower the top plane (which produces the inner chain). Thus we would obtain two quadrilaterals without multiple hinges per vertex, and it would be clear which edge to cut. In fact the resulting shape would be quite similar to that given in Figure 7.2.

The arguments above apply to any vertex of an inner polygon that has multiple hinges. Therefore any such vertex has a hinge that can be cut safely. Precise details for the non-nested case are not mentioned here, but by the limit argument mentioned above, we conclude the following:

Theorem 7.17. *All* closed *bands may be unfolded.*

7.7 Unfolding prismatoids

We have shown that arbitrary bands may be unfolded without overlap. It remains interesting to determine if this can lead to an unfolding of prismatoids without overlap, including the top and bottom faces, by cutting one hinge and all but one edge on each of the parallel planes. It is natural to hope that the top and bottom faces could be placed flat on opposite sides of the unfolded band, but it is not obvious how to ensure non-overlap.

It seems intuitive that prismatoids which produce nested bands can be unfolded, by cutting *all* hinges and all but one of the edges on the outer polygon. An interesting question is whether such cuttings extend to arbitrary bands.

7.8 Cutting a geodesic: O'Rourke's conjecture

As mentioned after Lemma 7.6, for nested bands it is easy to see that a geodesic cut is safe. First, this shortest path must be orthogonal to each slice curve, otherwise it could be adjusted to be made shorter. Second, the shortest path occurs on the trapezoid with steepest slope, and in fact such a trapezoid will resemble the one in Figure 7.11. Thus the shortest path passes through only one trapezoid in a nested band, so we can insert an orthogonal hinge and use the proofs for nested bands to show that no overlap will occur. In general, even if the shortest cut is a polygonal path C that passes through more than one trapezoid, the fact that it is geodesic implies that in the flattened state C will be straight. Thus, once flat, C will resemble a hinge that is orthogonal to its adjacent edges. The arguments given in this chapter prove that the resulting configuration has no overlap.

Chapter 8

Conclusion

The main contributions of this thesis are the following:

- The minimum area between two orthogonal monotone chains, over all translations, can be computed in $O(n^2 \log n)$ time.

- Reconfiguration of geometric triangulations can be achieved with $O(n \log n)$ edge flips and point moves.

- Flat chains with unit-length edges and each angle within the range $(60°, 150°)$ are connected with dihedral motions.

- There exist at least nine distinct classes of hexagonal unknots in the universal model of motion.

- All polyhedral bands can be cut along one edge and unfolded.

This research has left certain open problems, some of which are listed here:

Open Problem 8.1. *What is the complexity of computing the minimum area between two monotone chains, over all possible translations?*

Open Problem 8.2. *Is it possible to reconfigure triangulations with $O(n)$ edge flips and point moves?*

Open Problem 8.3. *For appropriate edge lengths, do there exist more than four classes of hexagonal trefoils?*

Open Problem 8.4. *For the dihedral model, can the angle restrictions on unit-length chains be relaxed?*

It is difficult to allow angles to approach π, since this would simulate chains with arbitrary edge lengths in the dihedral model.

Open Problem 8.5. *Can band unfolding be applied to the unfolding of specific polyhedra?*

Bibliography

[AAC$^+$96] Oswin Aichholzer, Franz Aurenhammer, Siu-Wing Cheng, Naoki Katoh, Michael Taschwer, Günter Rote, and Yin-Feng Xu. Triangulations intersect nicely. *Discrete and Computational Geometry*, 16:339–359, 1996.

[AAH02a] Oswin Aichholzer, Lyuba Alboul, and Ferran Hurtado. On flips in polyhedral surfaces. *International Journal of Foundations of Computer Science*, 13(2):303–311, 2002.

[AAH02b] Oswin Aichholzer, Franz Aurenhammer, and Ferran Hurtado. Sequences of spanning trees and a fixed tree theorem. *Computational Geometry*, 21(1):3–20, 2002.

[AAHK03] Oswin Aichholzer, Franz Aurenhammer, Ferran Hurtado, and Hannes Krasser. Towards compatible triangulations. *Theoretical Computer Science*, 296:3–13, 2003.

[AAOS97] Pankaj Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine Schevon. Star unfolding of a polytope with applications. *SIAM Journal of Computing*, 26(6):1689–1713, 1997.

[ABC$^+$00] Hee-Kap Ahn, Prosenjit Bose, Jurek Czyzowicz, Nicolas Hanusse, Evangelos Kranakis, and Pat Morin. Flipping your lid. *Geombinatorics*, 10(2):57–63, 2000.

[ABD$^+$02] Greg Aloupis, Prosenjit Bose, Erik D. Demaine, Stefan Langerman, Henk Meijer, Mark Overmars, and Godfried Toussaint. Computing signed permutations of polygons. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 68–71 (extended version available in electronic proceedings), August 2002.

[ABD$^+$03] Oswin Aichholzer, David Bremner, Erik D. Demaine, Henk Meijer, Vera Sacristán, and Michael Soss. Long proteins with unique optimal foldings in the H-P model. *Computational Geometry: Theory and Applications*, 25(1–2):139–159, 2003.

[ABG$^+$04] Manuel Abellanas, Prosenjit Bose, Alfredo Garcia, Ferran Hurtado, Pedro Ramos, Eduardo Rivera-Campo, and Javier Tejel. On local transformations in plane geometric graphs embedded on small grids. In *Proceedings of the International Workshop on Computational Geometry and Applications (CGA)*, volume 2, pages 22–31, 2004.

[ABM04] Greg Aloupis, Prosenjit Bose, and Pat Morin. Reconfiguring triangulations with edge flips and point moves. In *Proceedings of the 12th International Symposium on Graph Drawing, Lecture Notes in Computer Science 3383*, pages 1–11, 2004. Also invited and submitted to Algorithmica.

[ACD$^+$02] Oswin Aichholzer, Carmen Cortes, Erik D. Demaine, Vida Dujmovic, Jeff Erickson, Henk Meijer, Mark Overmars, Belen Palop, Suneeta Ramaswami, and Godfried Toussaint. Flipturning polygons. *Discrete and Computational Geometry*, 28:231–253, 2002.

[ACH$^+$91] Esther Arkin, Paul Chew, Daniel Huttenlocher, Klara Kedem, and Joseph Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, March 1991.

[ADD$^+$02] Greg Aloupis, Erik D. Demaine, Vida Dujmovic, Jeff Erickson, Stefan Langerman, Henk Meijer, Joseph O'Rourke, Mark Overmars, Michael Soss, Ileana Streinu, and Godfried Toussaint. Flat-state connectivity of linkages under dihedral motions. In *Proceedings of the 13th Annual International Symposium on Algorithms and Computation, volume 2518 of LNCS*, pages 369–380, November 2002.

[ADE$^+$01] Oswin Aichholzer, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael A. Soss, and Godfried T. Toussaint. Reconfiguring convex polygons. *Computational Geometry: Theory and Applications*, 20(1–2):85–95, October 2001.

[ADL$^+$04] Greg Aloupis, Erik Demaine, Stefan Langerman, Pat Morin, Joseph O'Rourke, Ileana Steinu, and Godfried Toussaint. Unfolding polyhedral bands. In *Proc. 16th Canadian Conference on Computational Geometry*, pages 60–63, 2004. Extended version invited and submitted to Computational Geometry: Theory and Applications, with title Edge-Unfolding Nested Polyhedral Bands.

[ADM$^+$02] Greg Aloupis, Erik D. Demaine, Henk Meijer, Joseph O'Rourke, Ileana Streinu, and Godfried Toussaint. On flat-state connectivity of chains with fixed acute angles. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 27–30, August 2002.

[AET04] Greg Aloupis, Günter Ewald, and Godfried Toussaint. More classes of stuck unknotted hexagons. *Beiträge zur Algebra und Geometrie - Contributions to Algebra and Geometry*, 45(2):429–434, 2004.

[AF96] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Math.*, 65:21–46, 1996.

[AFL$^+$03] Greg Aloupis, Thomas Fevens, Stefan Langerman, Tomomi Matsui, Antonio Mesa, Yurai Nuñez, David Rappaport, and Godfried Toussaint. Computing a geometric measure of the similarity between two melodies. In *Proceedings of the 15th Canadian Conference on Computational Geometry*, pages 81–84, Dalhousie University, Halifax, 2003. Extended version submitted to Computer Music Journal.

[AFM03] Esther Arkin, Sandor Fekete, and Joseph Mitchell. An algorithmic study of manufacturing paperclips and other folded structures. *Computational Geometry: Theory and Applications*, 25:117–138, 2003.

[AGP99] Boris Aronov, Jacob E. Goodman, and Richard Pollack. Convexification of planar polygons in $\mathbb{R}^3$. Manuscript (URL:http://cis.poly.edu/ aronov/research.html, October 1999.

[AKRW04] Helmut Alt, Christian Knauer, Günter Rote, and Sue Whitesides. On the complexity of the linkage reconfiguration problem. In Janos Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342, pages 1–14. American Mathematical Society, Contemporary Mathematics series, 2004.

[AKTvD00] Lyuba Alboul, Gertjan Kloosterman, Cornelis Traas, and Ruud van Damme. Best data-dependent triangulations. *Journal of computational and applied mathematics*, 119:1–12, 2000.

[AO92] Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete and computational geometry*, 8:219–250, 1992.

[AR05] Oswin Aichholzer and Klaus Reinhardt. A quadratic distance bound on sliding between crossing-free spanning trees. *Computational Geometry: Theory and Applications (invited). Also In Proc. 20th European Workshop on Computational Geometry, p.13-16*, 2005.

[AZ99] Martin Aigner and Günter Ziegler. *Proofs from THE BOOK*. Springer, Berlin, 1999.

[Bal03] Brad Ballinger. *Length-preserving transformations on polygons.* PhD thesis, University of California, Davis, 2003.

[BCG$^+$03] Prosenjit Bose, Jurek Czyzowicz, Zhicheng Gao, Pat Morin, and David R. Wood. Parallel diagonal flips in plane triangulations. Technical Report TR-2003-05, School of Computer Science, Carleton University, Ottawa, Canada, 2003.

[BDD$^+$98] Therese Biedl, Erik Demaine, Martin Demaine, Anna Lubiw, Mark Overmars, Joseph O'Rourke, Steve Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, pages 70–71, Montreal, Quebec, Canada, 1998.

[BDD$^+$01] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Mark Overmars, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. Locked and unlocked polygonal chains in 3D. *Discrete and Computational Geometry*, 26(3):269–281, 2001.

[BDD$^+$02] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. A note on reconfiguring tree linkages: Trees can lock. *Discrete Applied Mathematics*, 117:293–297, 2002.

[BDE$^+$03] Marshall Bern, Erik D. Demaine, David Eppstein, Eric Kuo, Andrea Mantler, and Jack Snoeyink. Ununfoldable polyhedra with convex faces. *Computational Geometry: Theory and Applications*, 24(2):51–62, 2003.

[BDL$^+$99] Therese Biedl, Erik Demaine, Sylvain Lazard, Steven Robbins, and Michael Soss. Convexifying monotone polygons. In *Proceedings of the Tenth International Symposium on Algorithms and Computation*, pages 415–424, Chennai, India, December 1999.

[BGRT99] Prosenjit Bose, Francisco Gómez, Pedro A. Ramos, and Godfried T. Toussaint. Drawing nice projections of objects in space. *Journal of Visual Communication and Image Representation*, 10:155–172, 1999.

[BHYJ88] B.K.Choi, H.Y.Shin, Y.I.Yoon, and J.W.Lee. Triangulations of scattered data in 3D space. *Computer Aided Design*, 20(5):239–248, 1988.

[Bie00] Therese Biedl. Polygons needing many flipturns. Technical Report CS-2000-04, Department of Computer Science, University of Waterloo (To appear in Discrete and Computational Geometry), January 2000.

[BNMW$^+$99] David Bainbridge, Craig G. Nevill-Manning, Ian H. Witten, Lloyd A. Smith, and Rodger J. McNab. Towards a digital library of popular music. In *Proceedings of the Fourth ACM International Conference on Digital Libraries*, 1999.

[BR95] Prabir Bhattacharya and Azriel Rosenfeld. Polygonal ribbons in two and three dimensions. *Pattern Recognition*, 28(5):769–779, 1995.

[BS05] Ciprian Borcea and Ileana Streinu. Singularities of hinge structures. *Manuscript*, 2005.

[Cal01] Jorge A. Calvo. The embedding space of hexagonal knots. *Topology and its Applications*, 112(2):137–174, 2001.

[Cau13] Augustin L. Cauchy. Sur les polygones et les polyèdres, seconde mémoire. *Journal de l'École Polytechnique*, 16(9):26–38, 1813.

[CCC$^+$00] Arbee L.P. Chen, Maggie Chang, Jesse Chen, Jia-Lien Hsu, Chih-How Hsu, and Spot Y.S. Hua. Query by music segments: An efficient approach for song retrieval. In *Proc. IEEE International Conference on Multimedia and EXPO (II)*, pages 873–876, 2000.

[CDIO04] Jason Cantarella, Erik D. Demaine, Hayley Iben, and James O'Brien. An energy-driven approach to linkage unfolding. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry (SoCG 2004)*, pages 134–143, Brooklyn, New York, June 2004.

[CDR02a] Robert Connelly, Erik D. Demaine, and Günter Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In Jorge Calvo, Kenneth Millett, and Eric Rawdon, editors, *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in* $\mathbb{R}^3$, pages 287–311. American Mathematical Society, 2002.

[CDR02b] Robert Connelly, Erik D. Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete and Computational Geometry*, 30:205–239, 2002.

[CGH$^+$03] Carmen Cortés, Clara Grima, Ferran Hurtado, Alberto Marquez, Francisco Santos, and Jesus Valenzuela. Transforming triangulations of polygons on non planar surfaces. *Preprint submitted to Elsevier Science.*

(Also in Proc. 17th European Workshop on Computational Geometry, pp.31-34,2001), 2003.

[CGLQ03] Timothy Chan, Alexander Golynski, Alejandro Lopez-Ortiz, and Claude-Guy Quimper. Curves of width one and the river shore problem. In *Proceedings of the 15th Canadian Conference on Computational Geometry*, pages 73–75, 2003.

[CGMN02] Carmen Cortés, Clara Grima, Alberto Marquez, and Atushiro Nakamoto. Diagonal flips in outer-triangulations on closed surfaces. *Discrete Math.*, 254(1-3):63–74, 2002.

[Cho45] Gustave Choquet. Variétés et corps convexes. In *Association Française pour l'Avancement des Sciences*, pages 91–93, Congrès de la Victoire, 1945.

[CJ98] Jason Cantarella and Heather Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *Journal of Knot Theory and its Ramifications*, 7:1027–1039, 1998.

[CKM$^+$01] Jorge Alberto Calvo, Danny Krizanc, Patrick Morin, Michael Soss, and Godfried Toussaint. Convexifying polygons with simple projections. *Information Processing Letters*, 80(2):81–86, 2001.

[CLRS01] Thomas Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein. *Introduction to Algorithms.* McGraw-Hill, September 2001.

[CO01] Roxana Cocan and Joseph O'Rourke. Polygonal chains cannot lock in 4D. *Computational Geometry*, 20:105–129, 2001.

[Con78] Robert Connelly. A counterexample to the rigidity conjecture for polyhedra. *Inst. Haut. Etud. Sci., Publ. Math.*, 47:333–338, 1978.

[CR61] H.M. Cundy and A.P. Rollett. *Mathematical Models*. Oxford University Press, 1961.

[Cri76] Francis H. C. Crick. Linking numbers and nucleosomes. *Proceedings of the National Academy of Sciences of the United States of America*, 73(8):2639–2643, 1976.

[Cro97] Peter R. Cromwell. *Polyhedra*. Cambridge University Press, 1997.

[Cro98] Charles Cronin. Concepts of melodic similarity in music-copyright infringement suits. In W.B. Hewlett and E. Selfridge-Field, editors, *Melodic Similarity: Concepts, procedures and applications*. MIT Press, Cambridge, Massachusetts, 1998.

[DDLO02] Erik D. Demaine, Martin L. Demaine, Anna Lubiw, and Joseph O'Rourke. Enumerating foldings and unfoldings between polygons and polytopes. *Graphs and Combinatorics*, 18(1):93–104, 2002.

[DDLS04] Erik D. Demaine, Martin L. Demaine, Jeffrey F. Lindy, and Diane L. Souvaine. Hinged dissection of polypolyhedra. In *Proceedings of the 14th Annual Fall Workshop on Computational Geometry*, pages 16–17, Cambridge, Massachusetts, November 19–20 2004.

[DEE$^+$02] Erik D. Demaine, David Eppstein, Jeff Erickson, George W.Hart, and Joseph O'Rourke. Vertex-unfolding of simplicial manifolds. In *Proceedings of the 18th annual symposium on computational geometry*, pages 238–243, Barcelona, Spain, 2002.

[DEJ97] Yuanan Diao, Claus Ernst, and Esias J. Janse van Rensburg. In search of a good polygonal knot energy. *Journal of Knot Theory and its Ramifications*, 6(5):633–657, 1997.

[Dem01] Erik D. Demaine. *Folding and Unfolding*. Ph.D. thesis, Department of Computer Science, University of Waterloo, Canada, 2001.

[Dew73] Alexander K. Dewdney. Wagner's theorem for torus graphs. *Discrete Mathematics*, 4:139–149, 1973.

[DLO03] Erik D. Demaine, Stefan Langerman, and Joseph O'Rourke. Geometric restrictions on producible polygonal protein chains. In *Proceedings of the 14th Annual International Symposium on Algorithms and Computation (ISAAC 2003)*, volume 2906 of *Lecture Notes in Computer Science*, pages 395–404, Kyoto, Japan, 2003.

[DLOS03] Erik D. Demaine, Stefan Langerman, Joseph O'Rourke, and Jack Snoeyink. Interlocked open and closed linkages with few joints. *Computational Geometry*, 26:37–45, 2003.

[DLR90] Nira Dyn, David Levin, and Samuel Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10:137–154, 1990.

[DOar] Erik D. Demaine and Joseph O'Rourke. A survey of folding and unfolding in computational geometry. In Jacob E. Goodman, János Pach, and Emo Welzl, editors, *Discrete and Computational Geometry*, Mathematical Sciences Research Institute Publications. Cambridge University Press, to appear.

[DORS88] Lester E. Dubins, Alon Orlitsky, Jim A. Reeds, and Lawrence A. Shepp. Self-avoiding random loops. *IEEE Transactions on Information Theory*, 34:1509–1516, 1988.

[Dür77] Albrecht Dürer. *The painter's manual: a manual of measurement of lines, areas, and solids by means of compass and ruler assembled by*

Albrecht Dürer for the use of all lovers of art with appropriate illustrations arranged to be printed in the year MDXXV. English translation of Unterweysung der Messung mit dem Zirkel un Richtscheyt in Linien Ebnen uhnd Gantzen Corporen, 1525. Abaris Books (Translated and with a commentary by Walter L.Strauss), New York, 1977.

[ELR$^+$98] Hazel Everett, Sylvain Lazard, Steve Robbins, Heiko Schröder, and Sue Whitesides. Convexifying star-shaped polygons. In *Proceeding of the Tenth Canadian Conference on Computational Geometry*, pages 2–3, 1998.

[EOS03] Jeff Erickson, Mark Overmars, and Michael Soss. Preprocessing chains for fast dihedral rotations is hard or even impossible. *Computational Geometry: Theory and Applications*, 26(3):235–246, 2003.

[Erd35] Paul Erdős. Problem number 3763. *American Mathematical Monthly*, 42:627, 1935.

[ES96] Herbert Edelsbrunner and Nimish R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.

[ET85] Hossam ElGindy and Godfried Toussaint. Efficient algorithms for inserting and deleting edges from triangulations. In *Proc. International Conference on Foundations of Data Organization*, Kyoto, Japan, 1985.

[FG99] Michael Floater and Craig Gotsman. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 101:117–129, 1999.

[FH76] Juan J. Freire and Arturo Horta. Mean reciprocal distances of short polymethylene chains. calculation of the translational diffusion coefficient of n-alkanes. *Journal of Chemical Physics*, 65:4049–4054, 1976.

[FHK$^+$96] Paul W. Finn, Dan Halperin, Lydia E. Kavraki, Jean-Claude Latombe, Rajeev Motwani, Christian Shelton, and Suresh Venkatasubramanian. Geometric manipulation of flexible ligands. In *Applied Computational Geometry*, pages 67–78. Springer-Verlag, 1996.

[FHM$^+$01] Thomas Fevens, Antonio Hernández, Antonio Mesa, Patrick Morin, Michael Soss, and Godfried Toussaint. Simple polygons with an infinite sequence of deflations. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 42(2):307–311, 2001.

[FK97] Maxim D. Frank-Kamenetskii. *Unravelling DNA*. Addison-Wesley, 1997.

[FNM00] Cristian Francu and Craig G. Nevill-Manning. Distance metrics and indexing strategies for a digital library of popular music. In *Proc. IEEE International Conference on Multimedia and EXPO (II)*, 2000.

[For93] Steven Fortune. A note on delaunay diagonal flips. *Pattern Recognition Letters*, 14:723–726, 1993.

[GCK91] Ulf Grenander, Yunshyong Chow, and Daniel M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer-Verlag, 1991.

[GHN$^+$03] Jerôme Galtier, Ferran Hurtado, Marc Noy, Stephane Pérennes, and Jorge Urrutia. Simultaneous edge flipping in triangulations. *Internat. J. Comput. Geom. Appl.*, 13(2):113–133, 2003.

[GHS00] Leonidas Guibas, John Hershberger, and Subhash Suri. Morphing simple polygons. *Discrete and Computational Geometry*, 24:1–34, 2000.

[GJ79] Michael R. Garey and David S. Johnson. *Computers And Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[GLCS95] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: Musical information retrieval in an audio database. In *ACM Multimedia*, pages 231–236, 1995.

[Grü95] Branko Grünbaum. How to convexify a polygon. *Geombinatorics*, 5:24–30, 1995.

[GS96] Wayne Goddard and Henda Swart. Distances between graphs under edge operations. *Discrete Mathematics*, 161:121–132, 1996.

[GS98] Christian Groß and Torsten-Karl Strempel. On generalizations of conics and on a generalization of the Fermat-Torricelli problem. *American Mathematical Monthly*, 105(8):732–743, 1998.

[GUW01] Zhicheng Gao, Jorge Urrutia, and Jianyu Wang. Diagonal flips in labelled planar triangulations. *Graphs Combin.*, 17(4):647–657, 2001.

[GY79] Lidija I. Golovina and Isaak M. Yaglom. *Induction in Geometry.* Mir Publishers, Moscow, 1979.

[GZ01] Branko Grünbaum and Joseph Zaks. Convexification of polygons by flips and flipturns. *Discrete Mathematics*, 241:333–342, 2001.

[HDF03] Robert A. Hearn, Erik D. Demaine, and Greg N. Frederickson. Hinged dissection of polygons is hard. In *Proceedings of the 15th Canadian Conference on Computational Geometry (CCCG 2003)*, pages 98–102, Halifax, Nova Scotia, Canada, August 11–13 2003.

[HE02] Ludger Hofmann-Engl. Melodic similarity - a conceptual framework. In *Proc. 2nd International Conference on Understanding and Creating Music*, Naples, 2002.

[HHH02] Carmen Hernando, Michael E. Houle, and Ferran Hurtado. On local transformation of polygons with visibility properties. *Theoretical Computer Science*, 289(2):919–937, 2002.

[HJW84] John Hopcroft, Deborah Joseph, and Sue Whitesides. Movement problems for 2-dimensional linkages. *SIAM Journal of Computing*, 13(3):610–629, August 1984.

[HN99] Ferran Hurtado and Marc Noy. Graph of triangulations of a convex polygon and tree of triangulations. *Comput. Geom.*, 13(3):179–188, 1999.

[HNU99] Ferran Hurtado, Marc Noy, and Jorge Urrutia. Flipping edges in triangulations. *Discrete and Computational Geometry*, 22(3):333–346, 1999.

[HOS96] Sabine Hanke, Thomas Ottman, and Sven Schuierer. The edge-flipping distance of triangulations. *J. Universal Computer Science*, 2(8):570–579, 1996.

[HSF98] Walter B. Hewlett and Eleanor Selfridge-Field, editors. *Melodic Similarity: Concepts, procedures and applications.* MIT Press, Cambridge, Massachusetts, 1998.

[JWM90] Esias J. Janse van Rensburg, Stuart G. Whittington, and Neal Madras. The pivot algorithm and polygons: results on the FCC lattice. *Journal of Physics A: Mathematical and General Physics*, 23:1589–1612, 1990.

[KM95] Michael Kapovich and John Millson. On the moduli space of polygons in the Euclidian plane. *Journal of Differential Geometry*, 42:133–164, 1995.

[Knu76] Donald Knuth. Big omicron and big omega and big theta. *ACM SIGACT News*, 2:18–43, 1976.

[Kom97] Hideo Komuro. The diagonal flips of triangulations on the sphere. *Yokohama Math. J.*, 44(2):115–122, 1997.

[Lal69] Moti Lal. Monte Carlo computer simulations of chain molecules. *Molecular Physics*, 17:57–64, 1969.

[Law72] Charles L. Lawson. Transforming triangulations. *Discrete Mathematics*, 3:365–372, 1972.

[Law77] Charles L. Lawson. Software for c_1 surface interpolation. In J. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, New York, 1977.

[Lem00] Kjell Lemström. *String Matching Techniques for Music Retrieval.* PhD thesis, University of Helsinki, Faculty of Science, Department of Computer Science, 2000.

[LO96] Anna Lubiw and Joseph O'Rourke. When can a polygon fold to a polytope? Technical Report 048, Smith College, 1996.

[LT04] Anna Lubiw and Luke Tanur. Pattern matching in polyphonic music as a weighted geometric translation problem. In *Proc. 5th International Conference on Music Information Retrieval*, pages 289–296, Barcelona, Spain, October 10-14 2004. Universitat Pompeu Fabra.

[LW95] William Lenhart and Sue Whitesides. Reconfiguring closed polygonal chains in Euclidean d-space. *Discrete and Computational Geometry*, 13:123–140, 1995.

[LYZ01] Lie Lu, Hong You, and Hong-Jiang Zhang. A new approach to query by humming in music retrieval. In *ICME2001*, pages 22–25, Tokyo, 2001.

[Mar01] Isabel C. Martinez. Contextual factors in the perceptual similarity of melodies. *The Online Contemporary Music Journal*, 7, 2001.

[Mei75] Gary H. Meisters. Polygons have ears. *American Mathematical Monthly*, 82:648–651, 1975.

[MF04] Daniel Müllensiefen and Klaus Frieler. Measuring melodic similarity: Human vs. algorithmic judgements. In *Proceedings of the Conference on Interdisciplinary Musicology*, Graz, Austria, April 2004.

[MHK99] Jong-Sik Mo, Chang Ho Han, and Yoo-Sung Kim. A melody-based similarity computation algorithm for musical information. In *1999 Workshop on Knowledge and Data Engineering Exchange*, page 114, 1999.

[Mil94] Kenneth Millett. Knotting of regular polygons in 3-space. *Journal of Knot Theory and its Ramifications*, 3:263–278, 1994.

[MJHS85] Bruce MacDonald, Naeem Jan, Douglas L. Hunter, and Michael O. Steinitz. Polymer conformations through wiggling. *Journal of Physics A: Mathematical and General Physics*, 18:2627–2631, 1985.

[Mor98] Robert Morris. Sets, scales, and rhythmic cycles: a classification of *talas* in Indian music. In *21st Annual Meeting of the Society for Music Theory*, Chapel Hill, NC, December 1998.

[MS88] Neal Madras and Alan D. Sokal. The pivot algorithm: a highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50(1):109–185, 1988.

[Nag39] Béla de Szőkefalvi Nagy. Solution of problem 3763. *American Mathematical Monthly*, 46:176–177, 1939.

[Ó98] Donncha S. Ó Maidín. A geometrical algorithm for melodic difference. *Computing in Musicology*, 11:65–72, 1998.

[O'R00] Joseph O'Rourke. Folding and unfolding in computational geometry. In *Discrete Comput. Geom.*, volume 1763 of *Lecture Notes Comput. Sci.*, pages 258–266. Springer-Verlag, 2000. Papers from the Japan Conf. Discrete Comput. Geom., Tokyo, Dec. 1998.

[O'R01] Joseph O'Rourke. An extension of Cauchy's arm lemma with application to curve development. In *Discrete Comput. Geom.*, volume 2098 of *Lecture Notes Comput. Sci.*, pages 280–291. Springer-Verlag, 2001. Papers from the Japan Conf. Discrete Comput. Geom., Tokyo, Nov. 2000.

[O'R03] Joseph O'Rourke. On the development of the intersection of a plane with a polytope. *Computational Geometry: Theory and Applications*, 24(1):3–10, 2003.

[OS89] Joseph O'Rourke and Catherine Schevon. On the development of closed convex curves on 3-polytopes. *Journal of Geometry*, 35:152–157, 1989.

[PS80] Wolfgang J. Paul and Janos Simon. Decision trees and random access machines. *Logic and Algorithmics, Monograph 30, L'Enseignement Mathematique*, 1980.

[PS85] Franco Preparata and Michael Shamos. *Computational Geometry. An Introduction.* Springer-Verlag, 1985.

[PT01] Janos Pach and Gabor Tardos. Untangling a polygon. In *Graph Drawing*, pages 154–161, 2001.

[Ran88a] Richard Randell. Conformation spaces of molecular rings. In *Studies in Physical and Theoretical Chemistry*, pages 141–156. Elsevier Science Publishers, Amsterdam, 1988.

[Ran88b] Richard Randell. A molecular conformation space. In *Studies in Physical and Theoretical Chemistry*, pages 125–140. Elsevier Science Publishers, Amsterdam, 1988.

[Ran94] Richard Randell. An elementary invariant of knots. *Journal of Knot Theory and its Ramifications*, pages 279–286, 1994.

[Rei78] Angelika Reiser. A linear selection algorithm for sets of elements with weights. *Information Processing Letters*, 7:159–162, 1978.

[Sal73] G. Thomas Sallee. Stretching chords of space curves. *Geometriae Dedicata*, 2:311–315, 1973.

[SG72] Steven D. Stellman and Paul J. Gans. Efficient computer simulation of polymer conformation. I. Geometric properties of the hard-sphere model. *Macromolecules*, 5(4):516–526, 1972.

[SG00] Vitaly Surazhsky and Craig Gotsman. Morphing planar triangulations and polygons using convex representations. In *16th European Workshop on Computational Geometry, Eilat, Israel*, pages 31–34, 2000.

[She75] Geoffrey C. Shephard. Convex polytopes with convex nets. *Math. Proc. Camb. Phil. Soc.*, 78:389–403, 1975.

[Sos01] Michael Soss. *Geometric and Computational Aspects of Molecular Reconfiguration*. PhD thesis, McGill University School of Computer Science, 2001.

[SR76] Ernst Steinitz and Hans Rademacher. *Vorlesungen über die Theorie der Polyeder unter Einschluss der Elemente der Topologie*. Springer, Berlin, 1976. Reprint der 1934 Auflage, Grundlehren der Mathematischen Wissenschaften, No. 41.

[ST01] Michael Soss and Godfried Toussaint. Geometric and computational aspects of polymer reconfiguration. *Journal of Mathematical Chemistry*, 27(4):303–318, 2001.

[ST02] Michael Soss and Godfried Toussaint. Convexifying polygons in 3D: a survey. *Contemporary Mathematics*, 304:269–285, 2002.

[Str00] Ileana Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proceedings of the Forty-First Annual Symposium on Foundations of Computer Science*, pages 443–453, Redondo Beach, California, November 2000.

[Str04] Ileana Streinu. Pseudo triangulations, rigidity and motion planning. *Discrete and Computational Geometry (submitted)*, 2004.

[SW88] De Witt L. Sumners and Stuart G. Whittington. Knots in self-avoiding walks. *J.Phys. A: Math.Gen.*, 21:1689–1694, 1988.

[SZ67] Isaac J. Schoenberg and S. K. Zaremba. On Cauchy's lemma concerning convex polygons. *Canadian Journal of Mathematics*, 19(4):1062–1071, 1967.

[TGV$^+$03] Rainer Typke, Panos Giannopoulos, Remco Veltkamp, Frans Wiering, and René van Oostrum. Using transportation distances for measuring melodic similarity. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 107–114, 2003.

[Tou99a] Godfried T. Toussaint. Computational polygonal entanglement theory. In *VIII Encuentros de Geometría Computacional*, pages 269–278, Castellon, Spain, July 1999.

[Tou99b] Godfried T. Toussaint. The Erdős-Nagy theorem and its ramifications. In *Proceedings of the Eleventh Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999.

[Tou01] Godfried T. Toussaint. A new class of stuck unknots in Pol_6. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 42(2):301–306, 2001.

[Tou03] Godfried T. Toussaint. Simple proofs of a geometric property of four-bar linkages. *American Mathematical Monthly*, 110(6):482–494, 2003.

[Tou04] Godfried T. Toussaint. A comparison of rhythmic similarity measures. In *Proc. 5th International Symposium on Music Information Retrieval*, pages 242–245, Barcelona, Spain, October 2004. Universitat Pompeu Fabra.

[ULM03] Esko Ukkonen, Kjell Lemström, and Vell Mäkinen. Geometric algorithms for transposition invariant content-based music retrieval. In *Proceedings of the International Conference on Music Information Retrieval ISMIR*, pages 193–199, 2003.

[Wag36] Klaus Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.

[Weg93] Bernd Wegner. Partial inflation of closed polygons in the plane. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 34(1):77–85, 1993.

[Whi01] Sue Whitesides. Chain Reconfiguration: the Ins and Outs, Ups and Downs of moving polygonal linkages. In P. Eades and T. Takaoka, editors, *Twelfth International Symposium on Algorithms and Computation*, volume 2223 of *LNCS*, pages 1–13. Springer-Verlag, 2001.