Bachelor's Thesis Computing Science

Radboud University Nijmegen

Polyhedra Folding Algorithms

A study on geometric folding algorithms and their practical applications

Author:
Sabin-Gabriel Iacob
s1084850

First supervisor/assessor:
Prof. Dr. Frits Vaandrager

Second assessor:
Dr. Lucia Cavallaro

July 11, 2024

Abstract

This thesis explores polyhedra folding algorithms, a lesser-known field of computational geometry. It begins by covering fundamental geometric concepts such as polygons, polyhedra, and convexity, followed by key aspects of computational geometry, including folding, unfolding, and curvature. The thesis then has three core research parts. In the first part, we provide a detailed analysis of methods for unfolding polyhedra. In the second part, we analyze folding methods for polygons, as well as a dynamic programming algorithm for a class of polygons. Lastly, in the third section, practical applications of folding/unfolding are examined in fields such as medicine, manufacturing, robotics, architecture, and origami design. This study aims to make advanced topics in geometric folding more accessible to third-year computer science students, highlighting their interdisciplinary relevance and potential for future research.

Contents

Chapter 1

Introduction

The research area of polygon folding lies at the intersection between mathematics and computer science [13, 44]. The mathematics brings familiar concepts from geometry, while computer science entails the algorithmic design. Works ranging from the mid 50's until late 90's focus mainly on geometry, while more recent works show the use of computers in computational problems. In [42] for example, a C++ program was written to find unfoldings of various shapes. Results range from the cube, which has 11 different non-overlaping unfoldings along the edges, to the Icosidodecahedron, with over 1 trillion of such unfoldings.

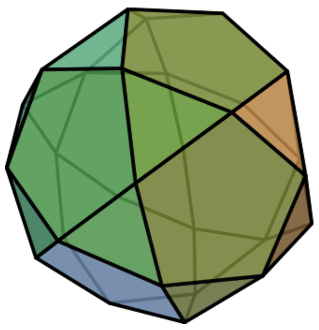


Figure 1: Icosidodecahedron [1]

Polygon folding, or Polyhedra folding, is the commonly used name of this field. It is a sub-field of *Computational geometry*, and it seeks answer questions such as "can this shape be folded out of this shape" or "in how many distinct ways can this box be unfolded" or "how many different boxes can be folded from the same polygon"[29]. These questions also explain the motivation behind this research area; its findings have a direct application in the real world, in cardboard box manufacturing, for example, where trying everything by hand is either not feasible or not worth the effort. A rigid

[1]https://en.wikipedia.org/wiki/Icosidodecahedron

mathematical foundation based on geometry, in combination with a digital simulation using efficient algorithms provides a solution.

A downside to this solution is that it requires a deep understanding of two rather difficult research fields. The aim of this paper is to demystify the intricacies of polygon and polyhedron folding, presenting complex theories and methods into an accessible narrative. It should serve as a foundational resource for novice undergraduate researchers interested in geometrical folds.

The motivation behind this goal stems from the current fragmented state of literature on polygon folding. Although numerous studies have extended the horizons of the field, to the best of our knowledge there exists no unified, less than 400-page-long work that assimilates these separate pieces into a cohesive whole. This incoherence presents a significant barrier, especially for new researchers.

In this thesis, the goal is to present in a clear manner "the tip of the iceberg" of polygon folding, by carefully picking relevant works throughout the history and explaining, in accessible words, their findings. After this introduction, we continue with a chapter on background knowledge (Chapter 2), which places the reader into the correct theoretical, but also historical, framework. The technical backbone of this paper is divided into three chapters. Chapter 3 dives deep into the matter of polyhedra unfolding, followed by Chapter 4 with polygon folding, presenting concepts and showing how these directly translate into an efficient algorithm for a class of polygons. The last part (Chapter 5) provides a high level overview of folding used in a more practical setting.

Chapter 2

Background Knowledge

2.1 Historical Origins

When one thinks of folding, the first thought that comes to mind is likely *origami*, where paper is folded into various shapes, such as animals or flowers[1]. However, other definitions of folding have been defined over the years. Since Euclid established the basics of polygonal geometry around 300 B.C. [9], including multi-dimensional shapes, scholars have been exploring the correlation between dimensions and the process of converting from one to another. Folding/unfolding was the intuitive answer to converting from a dimension to another.

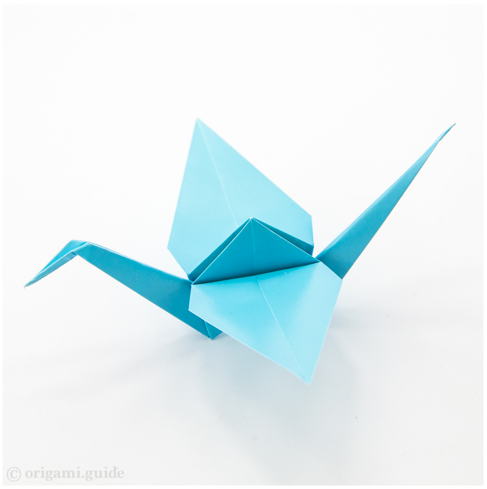


Figure 2: A crane, folded from one square sheet of paper[2]

It was not until the XVIth century, when the German painter Albrecht Dürer

[1] “ori” translates to “folding” and “kami” to “paper”, from Japanese; “kami” changes to “gami” due to Japanese morphology

[2] https://origami.guide/origami-animals/origami-birds/traditional-origami-crane/

(1471 - 1528) published a book, “Underweysung der Messung”[3] marking a pivotal moment in the understanding of geometrical structures, particularly two- and three-dimensional shapes [16]. Leading up to this publication, Dürer’s interest in geometrical shapes can be seen from some of his works (Figure 3, where a shape resembling a cube can be seen in the background).

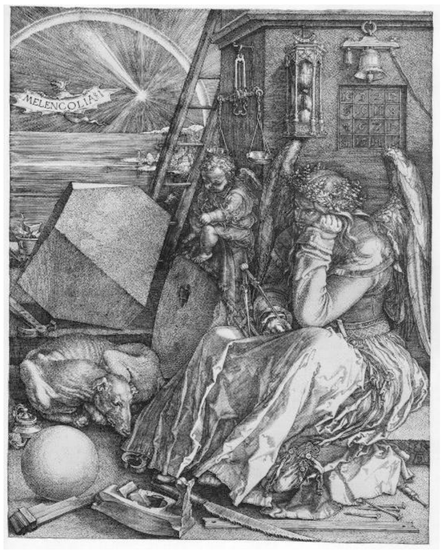


Figure 3: Melancholia I, 1515, Albrecht Dürer

Dürer’s illustrations, derived from his studies in Italy, depicted complex geometric shapes “opened” into 2D planes without overlapping, ensuring each shape remained connected and simple. These 2D polygons, named “nets” by the German author, laid the groundwork for the contemporary study of polygon folding, connecting the artistic side of folding to theoretical applications in mathematics (and later on, many other fields).

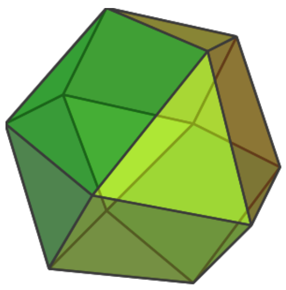


Figure 4: A cuboctahedron

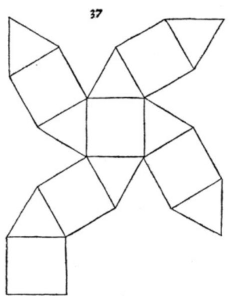


Figure 5: A cuboctahedron unfolded, as depicted by Albrecht Dürer

[3]Translated to “The Painter’s Manual”, from old German

This book, published in 1525, implicitly posed the question: *Does every convex polyhedron have a net?* This problem, formally articulated by Shephard in 1975 [41], has remained unsolved since then.

2.2 Geometry Concepts

Computational geometry is a field of computer science and mathematics, which includes folding and unfolding of polyhedra. Although no advanced concepts from computational geometry are used in this thesis, an understanding of advanced geometry is required. In this section, we will review key definitions that are assumed to be familiar to the reader. Readers may find it helpful to refer back to these definitions as needed while reading.

2.2.1 Polygon

A polygon is an enclosed planar shape consisting of straight segments, called edges. The points where these segments intersect are called vertices. A simple polygon's edges do not cross each other, and form vertices only at their endpoints. A polygon where edges cross each other is called complex.

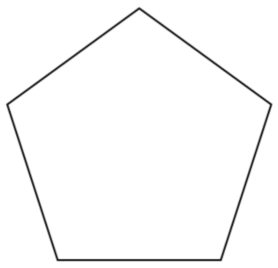


Figure 6: Simple polygon.

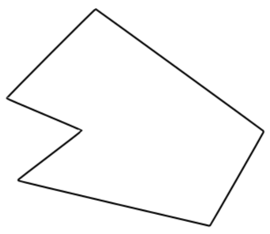


Figure 7: Simple polygon.

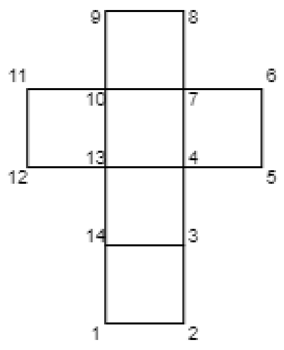


Figure 8: Complex polygon with 14 vertices and 19 edges.

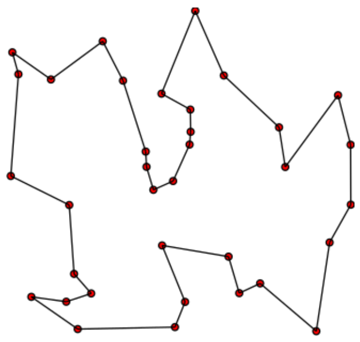


Figure 9: A polygon resembling a cyclic graph.

2.2.2 Geometric Notations

We will now go over notations which will be used throughout the thesis.

- L_P is the perimeter of polygon P (the sum of all edge lengths); it is a numerical value
- $|x, y|$ is the distance from a point x to a point y; it is a numerical value
- $\overline{xy}$ is the segment that has points x and y as endpoints; it is a physical element
- ∂P is the boundary of the polygon P (the continuous line made up of all the edges); it is a *physical* element
- $\alpha(v)$ is the angle of a vertex v; it is a numerical value expressed in orders of π, with $\pi = 180^\circ$; it is a numerical value

2.2.3 Polyhedron

A polyhedron (and plural, polyhedra) is a three-dimensional figure with polygons as faces. An edge of the polyhedron is the segment where two faces intersect. The points where the faces (or rather, edges of the faces) intersect are called vertices. These points are crucial in the unfolding process because "squishing" them is not always trivial. Formal explanations to this are given in the subsection *Curvature*.

There are infinitely many polyhedra. For example, the *platonic solids*, a class of polyhedra studied extensively by ancient Greek mathematicians, get their name from the philosopher *Plato*, who considered these shapes to have a cosmic meaning[4]. Figure 10 depicts the Platonic Solids[5].

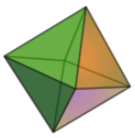
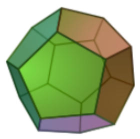
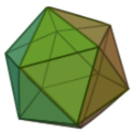
Tetrahedron	Cube	Octahedron	Dodecahedron	Icosahedron
Four faces	Six faces	Eight faces	Twelve faces	Twenty faces
				

Figure 10: Platonic Solids

[4]The dodecahedron, for example, Plato believed that "the god used [it] for arranging the constellations on the whole heaven". More information can be found in this article: https://www.cosmic-core.org/free/article-51-geometry-platonic-solids-part-12-the-dodecahedron/

[5]https://en.wikipedia.org/wiki/Platonic_solid

2.2.4 Convexity of Polygons and Polyhedra

Convexity is the property that if any two points within a polygon/polyhedra are chosen, the shortest path (line) connecting them lies entirely within the boundary. In other words, a convex polygon is topologically a circle, while a convex polyhedron is topologically a sphere. A non-convex shape is also called *concave*.

For example, the polygon from Figure 6 is convex, while the polygon from Figure 7, is not, because there exist two points, whose resulting segment partially lies beyond boundary (a "proof" can be seen in Figure 11). The platonic solids are convex polyehdra, while the *small stellated dodecahedron*[6] in Figure 12 is clearly not convex.

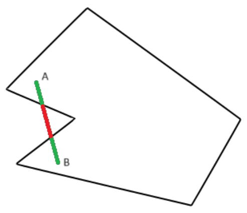


Figure 11: Non-convex polygon

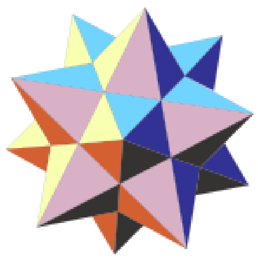


Figure 12: Non-convex polyhedron

2.2.5 Graph Theory

A tree is a connected graph with no cycles, and a spanning tree is a tree that includes all the vertices of a given graph.

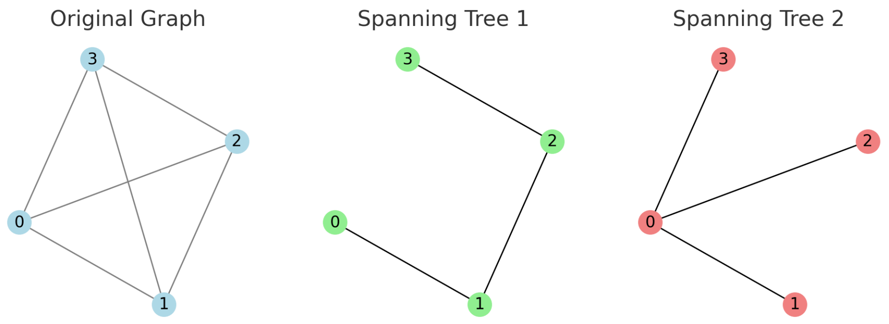


Figure 13: A graph and two spanning trees

2.2.6 Folding vs Unfolding

Folding and unfolding in the context of polyhedra involve transformations such that a two-dimensional (2D) surface is transformed into a three-dimensional

[6]https://en.wikipedia.org/wiki/Small_stellated_dodecahedron

shape (3D), and vice versa, without overlapping, preserving the connectivity (no disjoint sub-shapes).

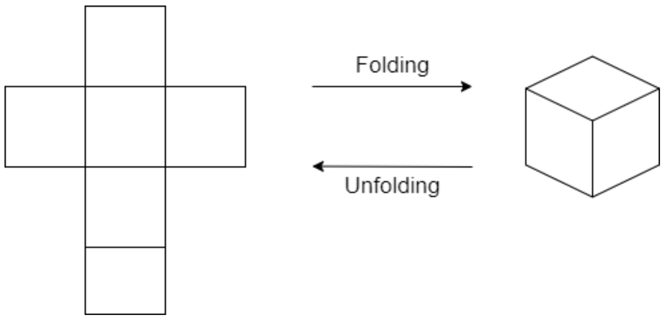


Figure 14: Cyclic relation of folding and unfolding

2.2.7 Nets

This term was coined by the German painter Albrecht Dürer and represents a single, non-overlapping, simply connected polygon resulting from unfolding a polyhedron. Note that the term is usually not used for a polygon which meets the criteria but is not the result of unfolding, or for a polygon for which it is not known whether it can fold into a polyhedron, without cuts or overlaps. In [16], Albrecht Dürer's unfolded the *Snub cube*[7] (Figure 15) into a shape resembling a star (Figure 16).

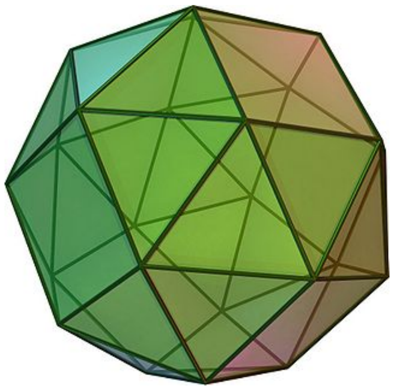


Figure 15: Snub cuboctahedron

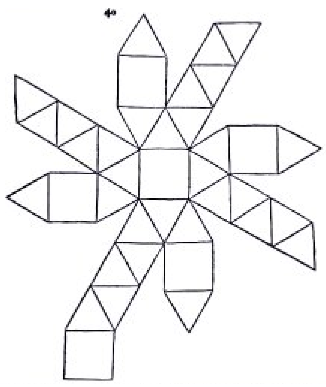


Figure 16: Snub cuboctahedron net

2.2.8 Curvature

Gaussian curvature, or simply curvature, represents one of the core concepts of advanced geometry. It originates from the differential geometry of smooth curves and surfaces, but it can also be adapted and used in polyhedra. René

[7]https://en.wikipedia.org/wiki/Snub_cube

Descartes (1596-1650) defines it as the *angle deficit* at a given point [19]: 2π minus the angle incident in that point.

On a polyhedron, curvature is relevant only at the vertices. A point on a face has 2π angle around it, thus curvature is zero. A point on an edge that is not an end-point has two angles of π incident, thus also curvature zero.

For example, all the vertices of a cube have a curvature of $\pi/2$ because at each vertex we have three connected squares, each with an angle of $\pi/2$, making the curvature be $2\pi - (\pi/2 + \pi/2 + \pi/2) = \pi/2$. If we add another vertex with curvature $\pi/2$ (e.g. a fourth square), the result will be a flat point with curvature zero. The curvature of a vertex v is denoted as $\kappa(v)$.

For the vertices of a polyhedron, we make the following distinction based on the curvature:

Zero Curvature: The vertex is lying on a face or an edge.

Positive Curvature: A cut is required to unfold the vertex.

Negative Curvature: Indicates a non-convex polyhedron and more than one cut is necessary.

Figure 17 [13] shows a non-convex polyhedron, where many faces concentrate in point v.

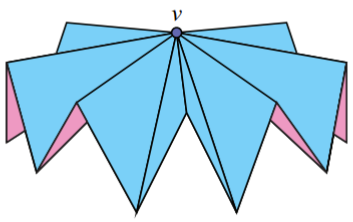


Figure 17: Point v has negative curvature

2.2.9 Cutting

Cutting is the “action” to unfold a polyhedron, sometimes also called “unfolding”. The lines where a 3D shape is opened represent the cuts made in the unfolding process. A valid cutting must visit all the non-zero curvature vertices. This implies that a cutting is a spanning tree of the graph that has all the vertices as nodes. Cutting negative curvature points is generally a challenge and will be discussed in Section 3.4.3.

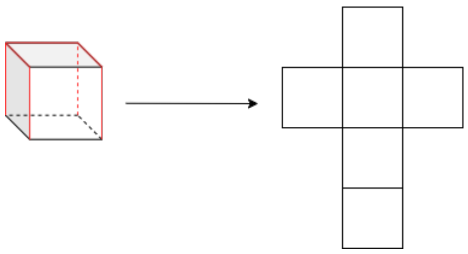


Figure 18: Cutting of a cube to result a Latin cross

2.2.10 Gluing

The opposite action of cutting. Gluing represents the action to "unify" two edges of a polygon, resulting in a 3D shape. In this thesis, the edges that are meant to be glued are connected by an arrow. In literature, *valley folds* and *mountain folds* are also commonly used.

A polygon with fold instructions marked with arrows is also called a gluing, and throughout this thesis we will use both meanings of gluing. Whether the polygon and the gluing actually result in a polyhedron will be discussed in Chapter 4 (section *Alexandrov Gluings*).

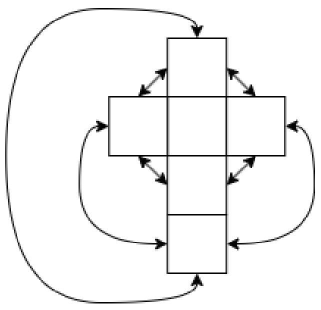


Figure 19: A gluing which results in a cube

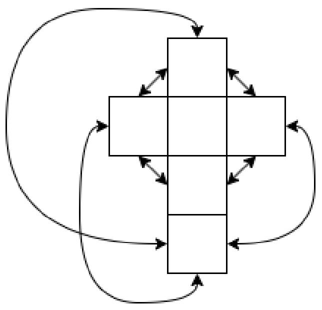


Figure 20: A gluing that does not result in a convex polyhedron

Chapter 3

Polyhedra Unfolding

3.1 Overview

The study of polyhedra unfolding involves the use of various techniques/methods to "flatten" three-dimensional shapes into two-dimensional representations without overlap. The lack of overlap sets this field apart from other geometrical folding paradigms, such as paper folding for example, where overlap is allowed (used in mathematical origami research[1]).

The significance of this research area comes from its theoretical and practical applications. For example, having a 2D representation of 3D shape helps with visualization, and is generally preferred in specific scenarios, such as architecture and manufacturing (more in Chapter 5). On the 2D gluing, shortest paths from the 3D model can be computed. This is relevant because we can define the a path using two coordinates, rather than three. For example, we know that the following gluing results in a cube. The shortest distance between vertices A and B is obvious on the cube (Figure 21), but on the gluing it does not even connect vertices A and B (Figure 22).

[1]This field has developed quite a lot in the last two decades. To our knowledge, the latest comprehensive work is "Introduction to Computational Origami"[44], and it is an impressive overview of both polygon and paper folding).

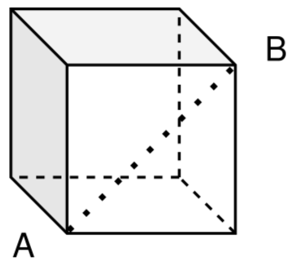


Figure 21: Shortest path between A and B on the cube

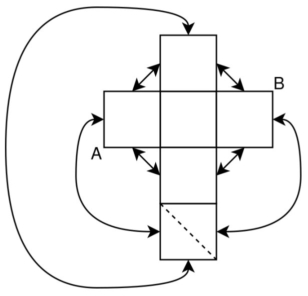


Figure 22: Shortest path between A and B on the unfolding

The concept of paths on 3D solids is relevant, for example, in plane flight planning [11]. A plane maintains constant altitude, thus its flight path is not a straight line, but rather a curved path, called geodesic [2]. Unfolding spherical shapes gave birth to a new class of algorithms: geodesic path planning. The research spans from its beginning in 1985, proposed by O'Rourke et al. [32], and is still ongoing at the time of writing, in 2024 [35].

In this chapter, the known unfolding methods will be discussed, then the folding problem will be examined, along with various attempts at solving it.

3.2 Edge Unfolding

Edge unfolding involves cutting a polyhedron along its edges to form a single, non-overlapping flat piece. This method is particularly significant in the manufacturing industry, where sheet metal parts are cut from flat sheets and bent into the desired shapes. The primary challenge with edge unfolding is ensuring that the unfolded piece does not overlap itself, a problem that remains unsolved for the class of convex polyhedra. Naturally, edge unfolding is bound by the number of edges.

3.3 General Unfolding

General unfolding allows cuts through the interior of the faces, not limited to edges. These methods guarantee that any convex polyhedron can be unfolded into a single, non-overlapping polygon. General unfolding is versatile and applicable to a wide range of polyhedra but is often computationally intensive.

[2]This article also explains this concept in a very intuitive and over-simplified way: https://gisgeography.com/great-circle-geodesic-line-shortest-flight-path/

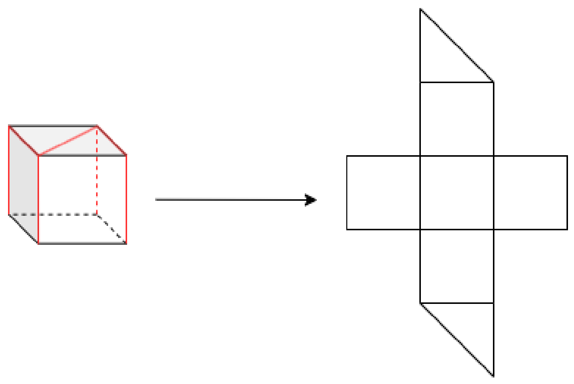


Figure 23: A face is cut along the diagonal

3.4 Unfolding Problem

The current status of the unfolding problem is summarized by the following table [13]:

	General unfolding	**Edge unfolding**
Convex polyhedra	Yes, various methods	Open problem
Non-Convex polyhedra	Open problem	No, counterexample

Although the first row has more interesting findings, we will tangentially address the challenges regarding non-convex polyhedra.

3.4.1 Convex Polyhedra

General Unfolding

The general unfolding of convex polyhedra is a problem that has been solved in many elegant ways throughout the years.

A lot of methods of general unfolding start from the *star of shortest paths*. The general idea is based on shortest geodesics from a chosen source x to the vertices of the shape. One property of such path is that it never crosses itself or another path[3]. Another property is that it never passes through a positive curvature vertex. An intuitive explanation is that it's always shorter to go around such a vertex. Additionally, they pass each face exactly once. These properties are key aspects in proving that the unfolded state is connected. The following unfolding techniques were based on these properties.

[3]One path can be a prefix or a sufix or another, but that case is an exception

Figure 24, from *Geometric Folding Algorithms* [13], depicts an example of such star of geodesics on a convex polyhedron with 100 vertices.

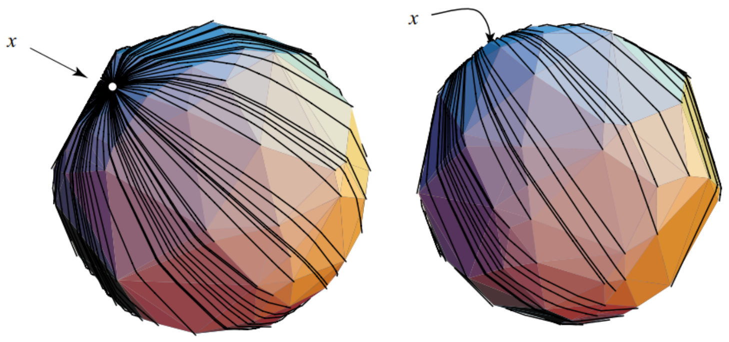


Figure 24: Paths spanning from x to all vertices

Star unfolding was first theoreticized by Alexandrov in 1955 [2], but the author claimed that it may overlap (*"Of course the polygon [...] may overlap itself when unfolded"*[3]). A proof that it does not overlap was published almost 40 years later, by Aranov et al.[5].

This folding can be applied if the source x is chosen such that all the shortest paths to the vertices are unique. If x meets this criteria, then cutting from each vertex along the edges of the graph unfolds without overlap. An example from [13] is shown bellow. The star is depicted by the continuous lines, going from x to all the vertices v_n.

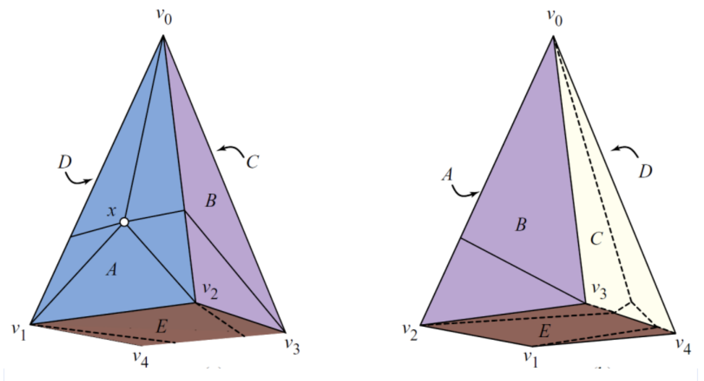


Figure 25: A pyramid

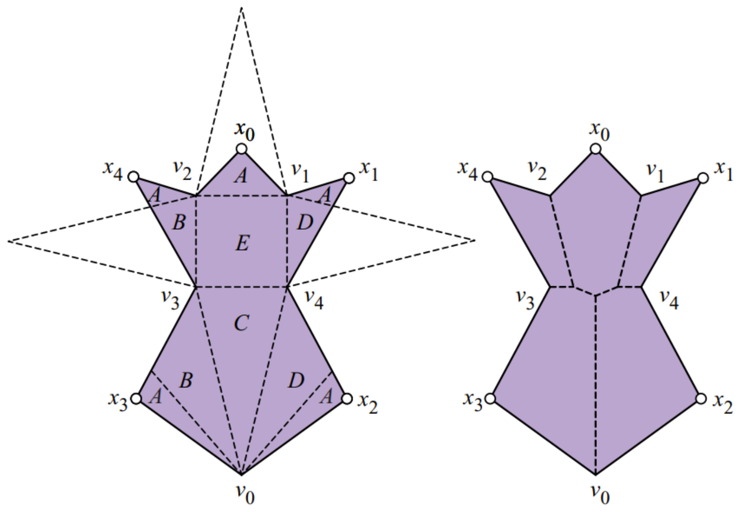


Figure 26: Star unfolding of the pyramid from Figure 25

Source unfolding was introduced in the computational community in the mid 80s [30, 39]. It relies in the the following lemma, based on Alexandrov's Theorem [3] (discussed in-depth in Chapter 4, section *Alexandrov's Theorem*):

Lemma. *Let p_1 and p_2 be distinct shortest paths emanating from x, which meet again at a point $y \in p_1 \cap p_2$ distinct from x. Then either one of the paths is a sub-path of the other, or neither p_1 nor p_2 can be extended past y while remaining a shortest path.*

In simpler terms, two of such paths cross at most once; a property that was already mentioned before-heand.

For this folding method, we can choose any source x. Once chosen, we draw the shortest path star. On any polyhedron, there will exist points that have a non-unique shortest path to x, and the set of these points forms the *cut locus*[23], or the *ridge tree*[39]. For consistency with graph notation, we will call it a *ridge tree.* This set of points can be found by leaving from x in two opposite directions and finding the intersection of these shortest paths. This intersection will be one of the nodes in the ridge tree, and the paths themselves will be the edges. There can be many variations of the ridge tree. Figure 25 depicts a ridge tree using the dotted lines on the "back" of the pyramid.

The cutting can be made along the edges of the tree described above. In contrast to the star unfolding, the source x stays central in the unfolding, with the initial star still visible.

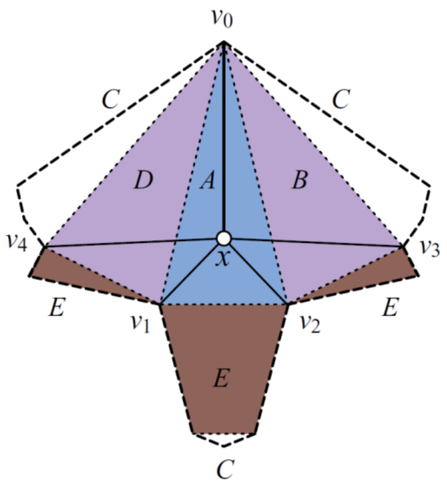


Figure 27: Source unfolding of the pyramid from Figure 25

Edge Unfolding

As seen in *Historical Background*, this problem was born in 1525 and has yet to be solved. In this thesis two techniques which have been used to unfold specific shapes are discussed.

Volcano unfolding is particularly suited for prismoids and dome-like structures. It involves cutting along edges that connect the base to the top, unfolding the sides around the base like a volcano, and flipping the top outward. This method has been proven to work without overlap for specific classes of polyhedra [13], but faces challenges when extended to more complex shapes, such as prismatoids. The figures bellow [13] depict examples of this unfolding.

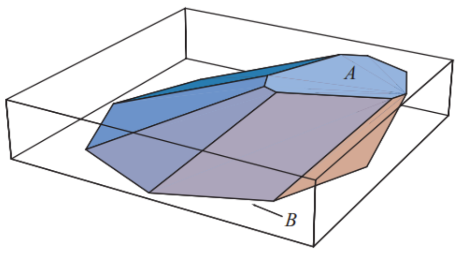


Figure 28: A prismoid

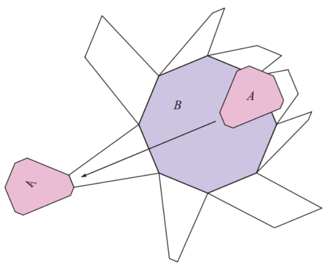


Figure 29: Volcano unfolding

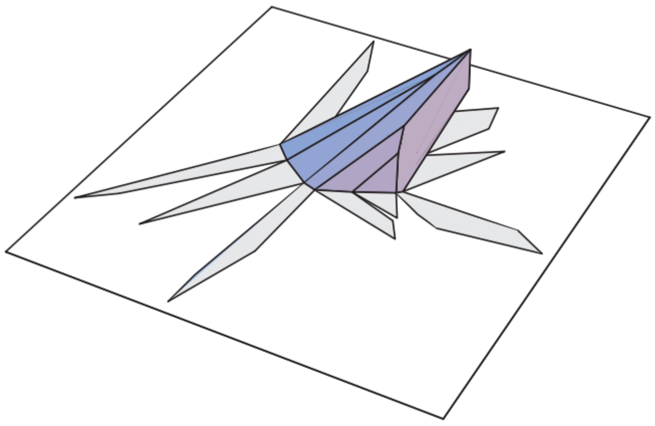


Figure 30: A dome

Belt unfolding is the opposite of Volcano Unfolding. It involves cutting along the edges that connect the top and the bottom faces to the lateral ones, keeping the lateral faces connected in a belt. The top and bottom are then kept attached to one of the belt faces. Both Volcano and Belt Unfolding still fail at very specific shapes (too many faces, or very wide bases) or in case the unfolding order is not correct.

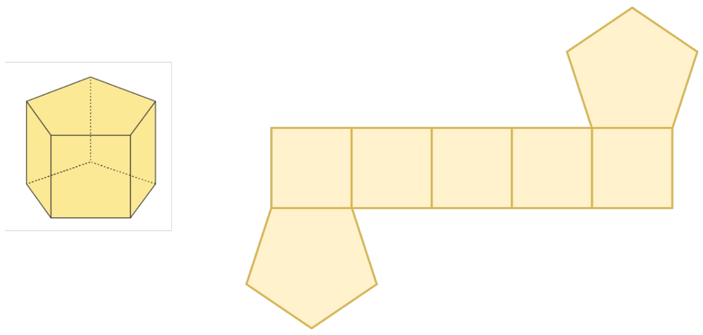


Figure 31: A pentagonal prism and its belt unfolding[4]

Related Work

Research is still in progress to find the net of every convex polyhedra, but the current results have not yet found an universal solution. A strong candidate for finding a solution was Schlickenrieder's hypothesis (1997), stating that a method called *Steepest Edge Unfolding* would find an unfolding for any convex polyhedra [37]. Figure 32 has been adapted from the companion slides of the undergraduate course based on [13].

[4]https://www.cuemath.com/geometry/pentagonal-prism/

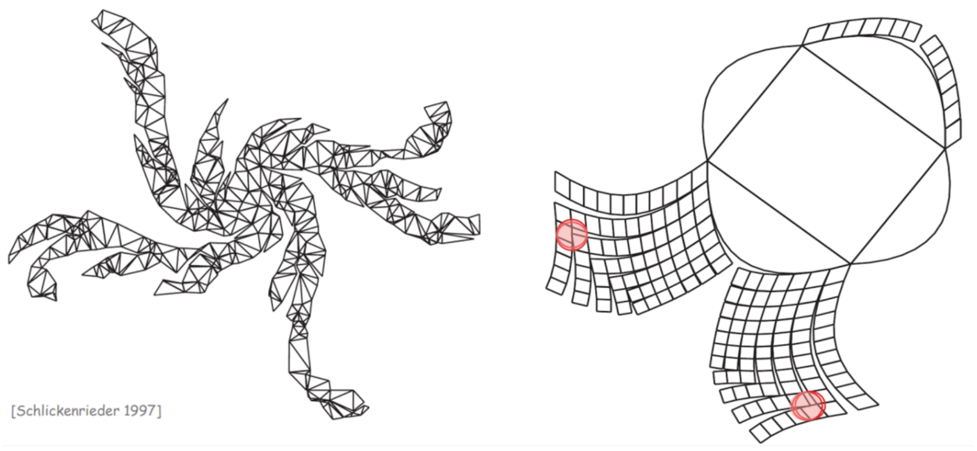


Figure 32: Steepest Edge unfolding results

Although the author himself found overlapping counterexamples to specific convex shapes (Figure 32, the net on the right), Schlickenrider's Master thesis concluded the following [37]:

> At each vertex, cut along the steepest ascending edge w.r.t. a certain objective function c. This approach does not always produce a net. However, experimenting with different (deterministic as well as randomized) choices of c, we found strong empirical evidence that for every polyhedron P there exists an objective function c such that this rule does produce a net.

This turned the 5-century old folding problem into an optimization problem: finding the best function c . In 2004, however, the claim was proved to be false [25].

Given this unfortunate situation, researchers have started seeking alternative problems to solve, which hopefully will contribute to solving the bigger problem once enough knowledge has been pooled. Such a problem was proposed by O'Rouke and Demaine in 2004 [13]. The problem allowed the edge unfolding to result in multiple disjoint polygons, aiming to minimize the their count. To our knowledge, the best result is 3/8 times the number of faces of the polyhedron [34].

Another sub-problem formulated involved finding overlapping edge-unfoldings. A recent paper addressing it presents an algorithm that computes all unfoldings for certain classes of (regular) convex polyhedra. The algorithm also counts how many of these unfoldings overlap [42].

The fact that not many papers are currently published on the topic also shows that research is very close to solving this problem. For example, recent result from December 2023 [33] uses the concept of the star of shortest paths and develops a technique resembling source unfolding to unfold a class of convex polyhedra.

In conclusion, we found various techniques which work for different classes of convex polyhedra, but no algorithm that englobes all convex shapes.

3.4.2 Non-Convex Polyhedra

In this subsection we will show counterexamples (for edge unfolding), and analyze an interesting technique which unfolds a class of non-convex polyhedra (for general unfolding). If finding an edge unfolding for convex polyhedra had an optimistic tone to it, finding any unfolding for non-convex polyhedra is close to impossible, due to the "irregularity" of these solids.

General Unfolding

As with the 5-century open problem, researchers have sought to solve restricted categories of shapes. In this thesis, we will only go over the findings on Orthogonal Polyhedra, particularly polycubes. Polycubes are a type of orthogonal polyhedra made of cubes of equal edge length, entirely glued together at the faces. The technique we will examine is called *grid unfolding.*

Grid unfolding is an interesting technique which unfolds various subclasses of Orthogonal Polyhedra, including polycubes. It first divides the surface into squares. In the literature, this technique is called *tiling.* A polycube is always depicted like this (to differentiate from a regular orthogonal polyhedra), so, depending on the context, this first step can be skipped.

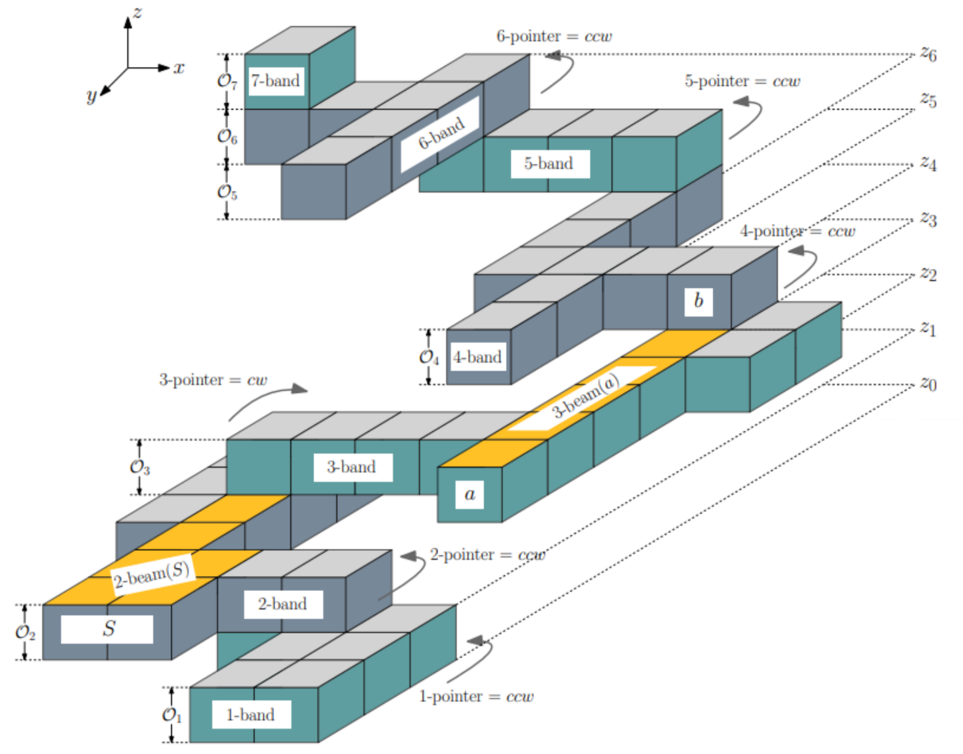


Figure 33: An polycube, divided into cubes

The different levels on the z-axis are called bands. The different levels on the y-axis are called beams. The levels on the x-axis are labeled as O_n and represent orthogonal faces. Damian and Meijer's paper [12] describes the correctness of grid unfolding. The polycube (Figure 33), as well as the unfolding (Figure 34) are adapted from their work. Note that the long "strips" are connected; top strip connects via L_3 to the middle strip, which is connected to the bottom strip via L_5.

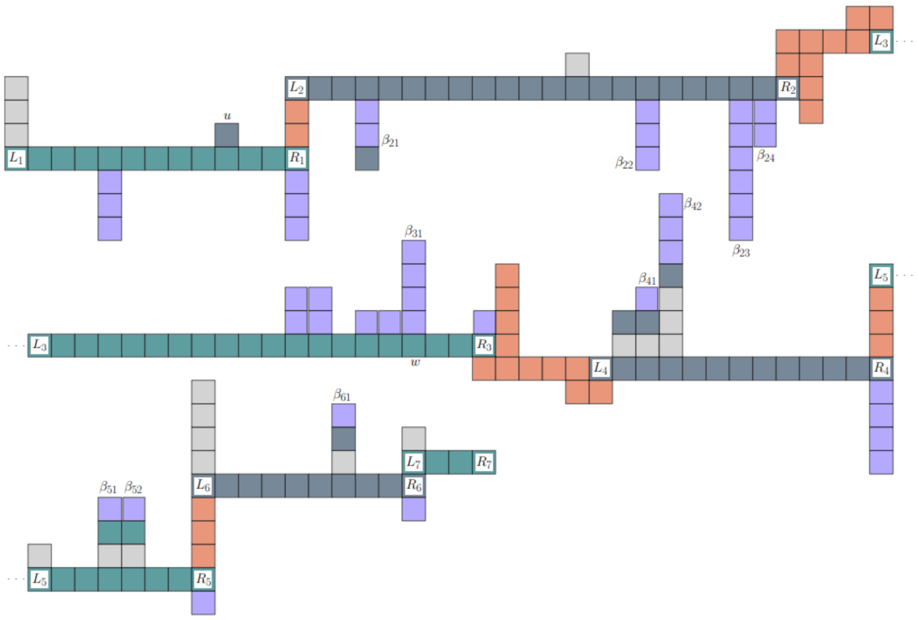


Figure 34: Grid unfolding of the polycube from Figure 33

Edge Unfolding

Unfortunately neither of the techniques used for convex polyhedra work for non-convex. An overly simplified explanation is that these techniques are well defined on convexity, and especially for positive curvature. A negative curvature vertex requires at least two cuts, and most of the time these cuts are going to discontinue the unfolding.

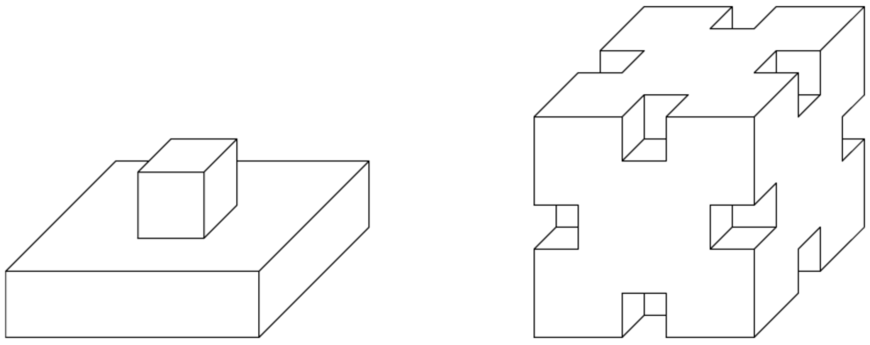


Figure 35: Orthogonal polyhedra that cannot be edge-unfolded [7]

Some non-convex polyhedra can be edge-unfolded though. A very recent paper (July 2024) shows a method based on *tabu search* [45]. The shapes it can unfold are truly impressive.

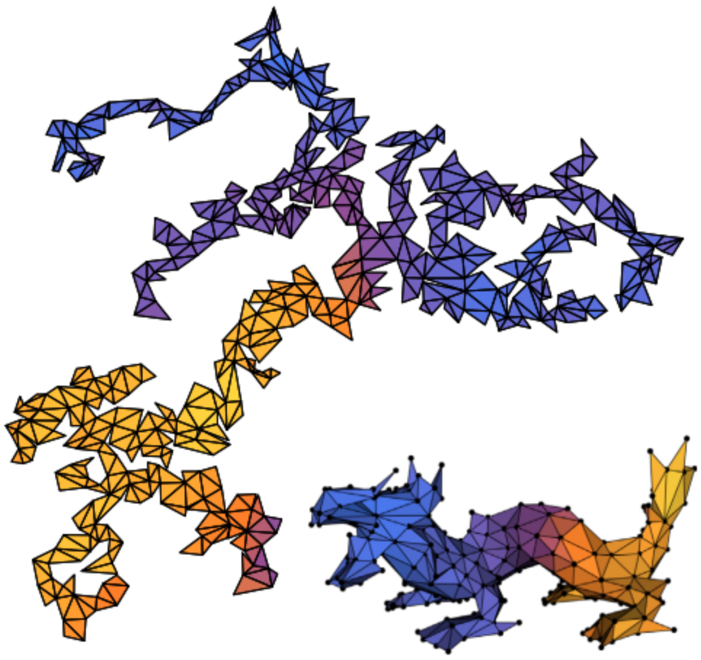


Figure 36: A dragon with 600 faces

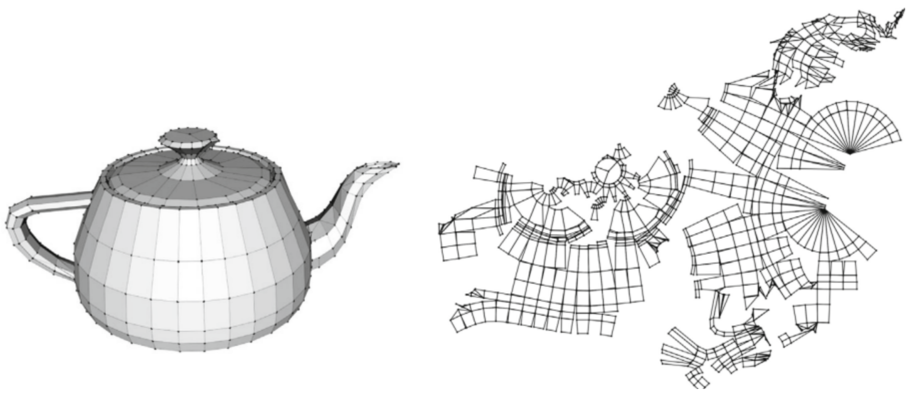


Figure 37: An Utah Teapot with 890 faces

This method it was only tested on a limited number of shapes with at most a few hundred faces. Another downside to this algorithm is that it cannot handle shapes which are not unfoldable. In other words, if the input is known to be unfoldable, the algorithm will efficiently ($O(n^3 logn)$) find its unfolding, otherwise it will fail.

Chapter 4

Polyhedra Folding

4.1 Overview

We have already established that unfolding polyhedra presents significant challenges. In contrast, the problem of folding polygons into polyhedra also encompasses considerable complexity.

In this chapter, we will first explore the theoretical foundations of folding, followed by an examination of the folding problem and the various techniques used to solve it.

4.2 Theoretical Foundation

Russian scientist and mathematician Aleksandr Danilovich Aleksandrov (1912-1999), whom we mentioned in Chapter 3, made significant contributions to the study of folding polyhedra in his 1955 work [2] (translated from Russian in 2005 [3]). Although Alexandrov's work was primarily focused on convex polyhedra rather than folding in particular, a variant of the folding problem was formally articulated by Lubiw and O'Rourke in 1996 [24], posed as *When can a polygon fold into a convex polyhedron?*,

4.2.1 Cauchy's Rigidity Theorem

We will first look at one of the theorems that laid the ground for mathematicians to follow. According to Cromwell [9], Cauchy formulated this work based on Euclide's work from 300 B.C., which contained a statement about comparing solids (polyhedra). This statement was modernly (at the time) formulated and proved by Cauchy in 1813, but Steinitz found a mistake in a lemma proof and corrected it in 1934 [43]. Additionally, Alexandrov extended this work to a polytope in any dimension in 1941 (previously limited to 3D), although in this thesis we are only analyzing 2D and 3D polytopes (polygons and polyhedra) [2].

A formulation of the theorem is as follows [13]:

Theorem. *If two convex polyhedra have the same combinatorial structure [...] and their corresponding faces are congruent [...], then the polyhedra are congruent [...]*

Combinatorial structure refers to the way in which the structure (faces, edges, vertices) of a polyhedron is arranged and connected (there can be many permutations of the structural elements). Two polyhedra have the same combinatorial structure, if there exists a surjective function that maps the elements from one polyhedron to the other such that the connections remain the same. Two faces are *congruent* if they can be superimposed through rigid transformations: translation, rotation and reflection. Similarly, two polyhedra are *congruent* if they can be perfectly overlapped through rigid transformations.

This theorem states that a 3D shape is dictated by the arrangement of its 2D elements. It serves as the backbone of proving that two polyhedra are the same.

4.2.2 Alexandrov Gluings

As mentioned in the preliminaries, we need a tool to check whether a gluing on a polygon can actually be folded. The key is a polyhedral metric. A *polyhedral metric* is a function that describes the distance between two points (metric) on a polyhedron (hence, polyhedral). It maps a pair of points to a geodesic. As the name suggests, a polyhedral metric operates on a polyhedron; however, a gluing can also be used to describe a polyhedron. Following this logic, a gluing is enough to define a distance function.

As with Cauchy's Rigidity Theorem, we aim to work on 2D whenever possible, rather than 3D. Figure 38 depicts an example of such metric, where the dotted line is the shortest distance between A and B, on the folded cube.

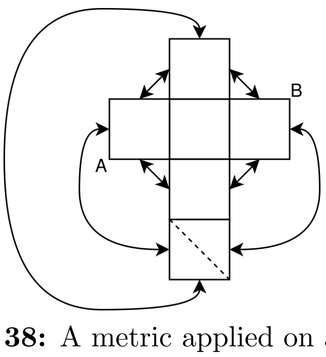


Figure 38: A metric applied on a gluing

Clearly, not all gluings can induce a polyhedral metric, i.e. are “valid”. In his work, *Convex polyhedra*, Alexandrov proved when a gluing can induce a metric, and this type of gluing was later named after him.

An *Alexandrov Gluing* is a gluing that meets following the conditions [3]:

> **Property 1:** The gluing should be convex (all points have positive curvature less or equal to 2π)
>
> **Property 2:** The gluing should be topologically a sphere (there should be no unglued edges and no crossings
>
> **Property 3:** The gluing should be polyhedral (finite number of points have non-zero curvature)

In other words, an Alexandrov gluing should not glue more than 360 degrees of “material” in one point, covers all the edges and the “arrows” don’t cross each other. Property 3 holds for all polygons that are finite and have no curved edges, which are essentially the all the polygons we discuss in this thesis.

4.2.3 Alexandrov’s Theorem

We have seen that Cauchy’s Rigidity Theorem (CRT) represents a uniqueness constraint. Alexandrov’s theorem (1941) builds on top of that, adding an existence claim [3]:

Theorem. *For every convex polyhedral metric, there exists a unique convex polyhedron realizing this metric.*

In the previous subsection, we have seen what a polyhedral metric is and how we can induce one using a gluing (convexity of the metric is guaranteed by the first property of an Alexandrov Gluing). Alexandrov’s Theorem basically tells us that an Alexandrov Gluing will result in a unique convex polyhedron.

4.3 Folding Problem

There are several version of this problem [13]:

> **Decision problem:** Given a polygon, can it fold into a convex polyhedron by gluing its edges?
>
> **Enumeration problem:** Given a polygon, list all the gluings that satisfy the Alexandrov Gluing.

Combinatorial problem: Given a polygon, into how many distinct convex polyhedra can it fold?

All of these problems have been extensively studied and solved (though there is room for optimization). Note that these problems target convex polyhedra simply because any polygon can (partially) fold into some non-convex shape. In this thesis, we will proceed with discussing the decision problem.

For convex polyhedra we have a straightforward answer for the decision problem: if the polygon has an Alexandrov Gluing, it will always fold into a unique convex polyhedron. The problem then turns into: *When does a polygon have an Alexandrov Gluing?* To address it, we will distinguish cases based on the convexity of the polygon.

4.3.1 Convex Polygons

All convex polygons can be folded into a convex polyhedron using the technique known as *Perimeter Halving* [13]. We will now discuss and prove this claim.

As the suggests, Perimeter halving halves the perimeter by choosing an arbitrary point $x \in \partial P$, then finding the point $y \in \partial P$ such that $|x, y| = |y, x| = L_P/2$, with $\overline{xy} \in \partial P$ and $\overline{yx} \in \partial P$. Figure 39 displays that.

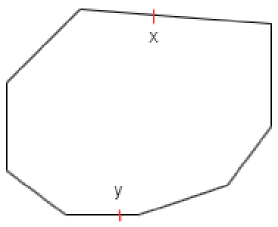


Figure 39: A convex polygon, with points x and y chosen as described

After this, starting from x, we go along ∂P in opposite directions, and whenever we encounter the vertex on either side we mark it (say v_1) and its symmetrical point over x on the opposite side (say w_1, resulting in $|v_1, x| = |x, w_1|$). Figure 40 depicts this first step. Now, in a recursive way, starting from the vertex which was first encountered (v_1 or w_1), repeat until v_n, w_n are found such that $|v_n, y| = |y, w_n|$. Figure 41 shows the second iteration of the algorithm.

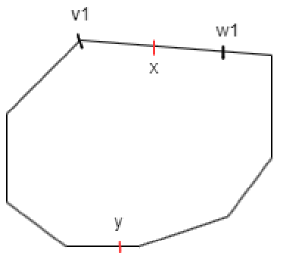


Figure 40: We find the first vertex v_1 and its symmetrical to x, which we mark as w_1

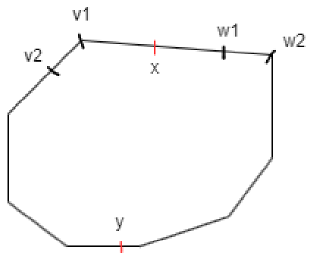


Figure 41: We find the next closest vertex on either side, w_2

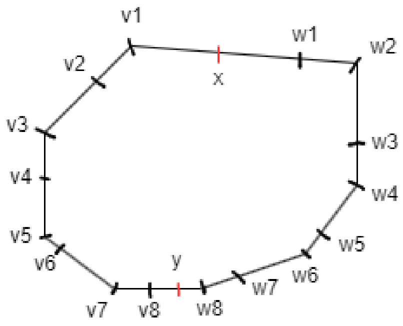


Figure 42: After marking all the references, we should end in y

As figure 42 depicts, we should now have ∂P divided into pairs of segments of equal lengths: $(\overline{xv_1}, \overline{xw_1}), (\overline{v_1v_2}, \overline{w_1w_2}), ..., (\overline{v_ny}, \overline{w_ny})$. We can now draw the gluing by connecting each segment with its pair (Figure 43), and we claim that it is an Alexandrov gluing, thus it folds into an convex polygon.

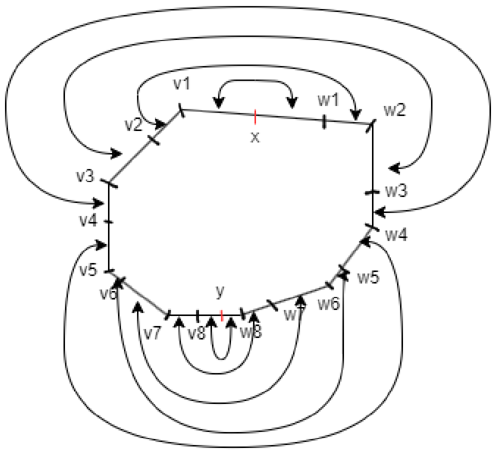


Figure 43: Alexandrov gluing resulting from Perimeter halving

Proof. Since the polygon is convex (property 3), all the vertices have an angle of strictly less than π, thus it is not possible to glue more than 2π material at any given time (property 1). Additionally, the entire boundary of the polygon is glued to itself and the pairs of segments are generated in such an order that there will be no overlaps during the gluing process (property 2). $\square$

In the case x and y are vertices, polygon gluing will still realize to a convex polyhedron, except for the case in which the other vertices are symmetrical to $\overline{xy}$ (note that $\overline{xy} \notin \partial P$). This will result in folding along $\overline{xy}$, halving the polygon, resulting in a perfect overlap. This is called a *degenerate case.* Although the outcome is a polygon, since it was obtained by folding, it is called a *degenerate polyhedron*[1]. For example, a square where x is one of the corners will result in a *doubly covered triangle.*

4.3.2 Non-Convex Polygons

The issue of folding non-convex polygons arises from vertices with an angle greater than π. In order to address this, we will make a (sub-)distinction on the edge lengths.

Matching edge-length polygons

We are going to analyze a specific class of polygons, which we called *matching edge-length* polygons. These are polygons where for every edge e_x there exists at least one edge e_y such that $|e_x| = |e_y|$. This makes gluing easier because we will only glue together edges of the same length (this is called edge-to-edge gluing); we only need to check that the curvature at the endpoints does not go over 2π.

Lubiw and O'Rourke proposed an algorithm that finds such gluings in their work from 1996 [24]. The algorithm solved the enumeration and combinatorial problem for matching edge-length polygons in exponential time; the same algorithm was then adapted to solve the decision problem in polynomial time. The algorithm is based on the principle of dynamic programming.

Dynamic programming refers to splitting the problem into smaller non-overlapping sub-problems then solving these, usually recursively, to solve the whole problem. In our context, the algorithm selects a pair of matching length edges, and if the edges can be glued then it proceeds to solving the two sub-polygons resulting from gluing two edges.

Figure 44 shows a polygon (left) where gluing that we know is Alexandrov.

[1]https://en.wikipedia.org/wiki/Degeneracy_(mathematics)

To help visualization, we are going to introduce a slightly different perspective on folding polygons. Instead of drawing a 3D solid, we are going to draw the two edges close to each other, showing that they have been glued. This is called a *gluing tree*, and its the main tool on which folding algorithms have been developed. *Gluing trees* have a lot of properties associated that govern the validity of an iteration of an algorithm, but in this thesis, we are only going to use them as a tool for visualization within the context of this dynamic programming algorithm. In the same figure, the gluing tree is also depicted (right). The red x-marks suggest the nodes in the gluing tree, which correspond to the vertices in the resulting polyhedron (cube).

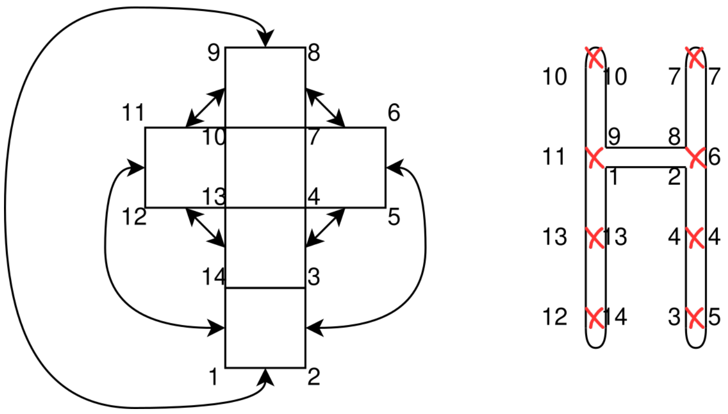


Figure 44: A gluing and its gluing tree

Figure 45 shows another example, where only edges $e1$ and $e2$ are glued. The curved lines represent the rest of the edges $\overline{v1v2}$, $\overline{v2, v3}$, $\overline{v3v4}$ and $\overline{v5v6}$, $\overline{v6v7}$, $\overline{v7v8}$, $\overline{v8v9}$ respectively. The points $v2, v3$ and $v6, v7, v8$ respectively are drawn somewhere on the curve respecting their topological order, but not necessarily their relative distance to each other. Edges $e1$ and $e2$ can now be seen as a "stitching", connecting two new sub-polygons: $P1$ and $P2$.

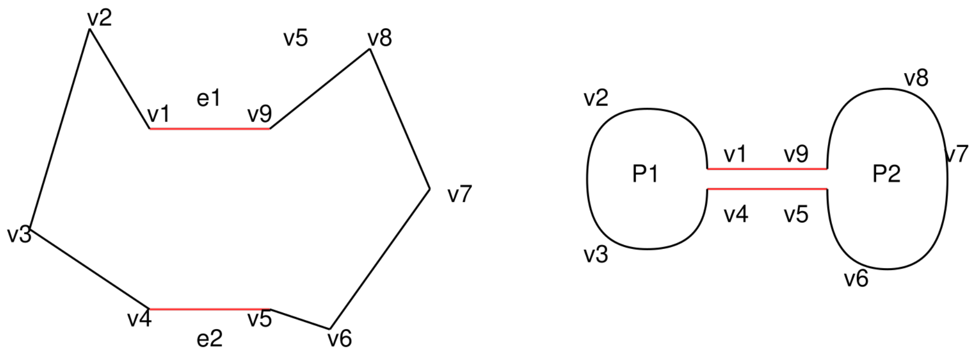


Figure 45: A gluing and its gluing tree

These two sub-polygons will be now be the sub-problems that the algorithm will proceed to solve. The end goal is to find an Alexandrov Gluing.

We will now formalize this algorithm and give its implementation in pseudocode. The algorithm and its description have been adapted from [13] and [24]. The pseudocode has been written as a companion to the description.

Dynamic programming algorithm for edge-to-edge gluing

We have a polygon P defined by the set of vertices $V_P = \{v_1, v_2, ..., v_n\}$ the set of edges $E_P = \{e_1, e_2, ..., e_n\}$ and the set of angles at a vertex $A_P = \{\alpha(1), \alpha(2), ..., \alpha(n)\}$, each angle corresponding to a vertex. An edge e_i has its endpoints in the vertices v_i and $v_{(i+1)\%n}$. The length of an edge is denoted as $l_i = |e_i|$. The polygonal chain, also called partial boundary, from v_i to v_j is denoted as $P[i,j]$. Obviously, $P[i,j] \in \partial(P)$. $P[i,j]$ can also be interpreted as the edges and the vertices, with the respective vertex angles, from v_i to v_j; this chain is closed (i.e. is a polygon itself) when $i = j$ or when v_i and v_j are glued. We denote as $\{e_i, e_j\}$ or $\{v_i, v_j\}$ when two edges, or two vertices respectively, are being glued to each other in a state of the gluing process.

The sub-problems are generated when two edges e_i and e_j are glued to each other. As we have seen before, this generated two sub-polygons, more precisely $P[i+1, j]$ and $P[j+1, i]$. An example can be seen in Figure 45, where edges e_9 and e_4 are glued, generating $P[1,4]$ and $P[5,9]$ (edge indexes have been adapted to reflect the notation of the algorithm, while vertex indexes have remain the same). The two sub-problems $P[i+1, j]$ and $P[j+1, i]$ are "linked" by $\{e_i, e_j\}$; an edge $e_x \in P[i+i, j]$ cannot be glued to an edge $e_y \in P(j+1, i)$, because $\{e_x, e_y\}$ would result in a shape that is not topologically a sphere. The main idea of the algorithm is to solve $P[i,j]$, for all valid combinations of i and j, and pick the "best" solution ("best" will be defined bellow). The sub-problem that coincides with the main problem is $P[1,1]$ (or any i such that $i = j$), and that is why the algorithm falls in the class of dynamic programming.

We can now turn our attention to a gluing of vertices, say v_i and v_j. We denote the resulting angle of the associated match $\{v_i, v_j\}$ as $\alpha(i,j)$, and the key aspect of the algorithm is that it tries to minimize the angle it glues at any point. The gluing which minimizes angles glued together is, using a greedy approach, the “best” because it is less likely to glue more than $360°$ at any given point. We will write that as $\alpha_{min}(i,j)$, we can draw the following definitions:

- **D1.** If $|j-i|$ is odd, one edge in $P[i,j]$ does not have a matching edge of equal length. Thus we define $\alpha_{min}(i,j) = \infty$, because the polygonal chain of i and j cannot be glued together
- **D2.** If $i = j$, then we are gluing a vertex to itself. This happens when we fold two adjacent edges. Trivially, the result angle is $\alpha_{min}(i,j) = 0$
- **D3** If $|j-i| \geq 2$ and even, then we have an even length polygonal chain where every edge could have a match. We now want to find a $k, i < k \leq j-1$ with a different parity[2] from i such that $\{e_k, e_i\}$ is valid. To do that, we try all values for k such that $\alpha_{min}(i,j)$ is minimized when adding the extra angle $\Delta(k)$ from gluing e_k. If $l_i \neq l_k$, then $\alpha_{min}(i,j) = \infty$. Otherwise, this creates two other sub-problems:
  - $P[v_{i+1}, v_k]$: If $i+1 = k$, this problem is vacuous and $\Delta(k)$ is determined by the second sub=problem. Assume $i+1 \neq k$. If $\alpha_{i+1} + \alpha(k) + \alpha(i+1,k) < 2\pi$, $\Delta(k)$ is determined by the second sub-problem, otherwise $\{e+k, e_i\}$ is not valid and $\alpha_{min}(i,j) = \infty$
  - $P[v_{k+1}, v_j]$: $\Delta(k) = 0$ if $k+1 = j$, otherwise $\Delta(k) = \alpha(k+1) + \alpha_{min}(k+1, j)$

Definition D3 is where the recursive calls are made, and to explain it better we will depict a small example using the polygon in Figure 46 .

[2]Optimization trick to avoid ending up in case D1

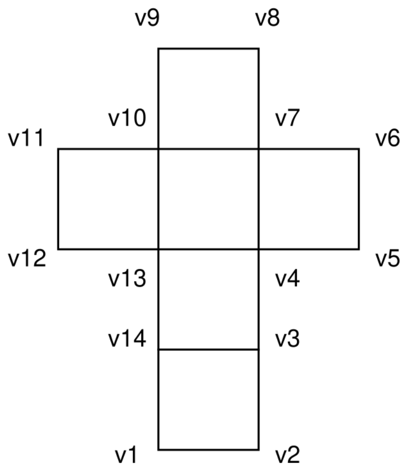


Figure 46: A complex polygon, the Latin Cross

Say we want to compute $\alpha_{min}(2,6))$ corresponding to $\{v_2, v_6\}$. There are two valid values for k: 3, 5. Case $k = 3$ corresponds to $\{e_2, e_3\}$. This yields $P[v_2, v_2]$ (D1), and $P[v_4, v_6]$, which result in the extra angle $\Delta(3) = \alpha(4) + \alpha_{min}(4,6) = 3/2\pi + 0 = 3/2\pi\,(270^{\circ})$. Case $k = 5$ corresponds to $\{e_2, e_5\}$ and generates two sub-problems ($P[v_3, v_4]$ and $P[v_6, v_6]$ (D1). The first sub-problem is legal since $\Delta(5) = \alpha(3) + \alpha(5) + \alpha_{min}(3,5) = \pi + 1/2\pi + 0 = 3/2\pi < \pi$. Finally, $\alpha_{min}(2,6) = min(\Delta(3), \Delta(5)$, meaning that the chain $P[2,6]$ can me folded such that no extra angle is matched to $v2 = v6$ (i.e. the final polyhedron, a cube, has angle, or curvature, 270° in that point).

The complexity of the algorithm is $O(n^3)$. This is explained using the pseudocode bellow:

```
// $O(n^2)$ subproblems
for each subproblem $P[i,j]$ do
    // $O(n)$ choices for $k$, then constant computations
    find $k$ such that it minimizes the angle glued at the point $i = j$;
end
```

Generic polygons

We will start off by pointing out that there exist polygons which cannot fold at all into a convex polyhedron. Figure 47 depicts such a polygon, where “inner” vertices have an angle of more than π. No matter how the polygon is glued, the result will have a point with curvature greater than 2π.

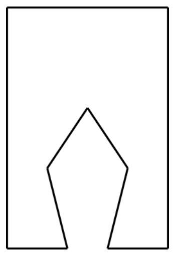


Figure 47: A non-convex polygon that cannot be glued

This raises the question that if we generate a random polygon (with any reasonable definition of random), what is the probability that it will fold into a convex polyhedron? With a random polygon with n vertices drawn from some continuous density distribution, at least half of the vertices will be concave. We will now claim and prove the following statements based on the number of such vertices.

The following lemma is derived from the definition of Perimeter Halving (although it was not explicitly labeled as a theorem)[13]:

Lemma. *A randomly generated polygon with exactly one vertex v with $\kappa(v) > \pi$ folds into a convex polyhedron.*

Proof. There exist three options for gluing a vertex:

i. Gluing v to a point v' on an edge then the resulting curvature will be $\kappa(v) + \kappa(v') > 2\pi$, which implies a non-Alexandrov gluing.

ii. If there is a Perimeter Halving configuration such that it glues v and v' with $\kappa(v') = 2\pi - \kappa(v)$, then it will result in a convex polyhedron. We will leave out this case, since the probability of having such two vertices is low in a randomly generated polygon.

iii. Gluing together the edges adjacent in a vertex is called *zipping*. If and only if Perimeter Halving either starts v, i.e. $x = v$, or ends in that point, i.e. x is chosen in such a way that $y = v$, the outcome is zipping at the vertex v. We will call these two edges $e1 = \overline{av}$ and $e2 = \overline{vb}$. If $|e1| = |e2|$, then the resulting end point c will have $\kappa(c) = \kappa(a)+\kappa(b) < 2\pi$, because $\kappa(a) < \pi$ and $\kappa(b) < \pi$. If $|e1| \neq |e2|$, then the end, say a with $\kappa(a) < \pi$ will glue to a point $a' \in e2$. Since $\kappa(a') = \pi$, this is an Alexandrov gluing.

□

We will now analyze the case where the polygon has more than two non-convex vertices (such as the one in Figure 47) [13]:

Lemma. *If a polygon has two or more concave vertices and random edge lengths, then there exists no Alexandrov gluing such that the polygon folds into a convex polygon.*

Proof. We consider a random polygon with two concave vertices $v1$ and $v2$. The convex vertices are denoted as $c_1, c_2, ..., c_k$. We will prove by induction on k, with the base cases of $k = 0$.
Case *k=0.* Very abstract, but can be imagined as two points being connected by two lines. Since the two edge lengths are generated by a continuous density function, they will have different lengths. The longer edge must be "pinched" and glued to the shorter one, resulting in a point of $2\pi - (\pi + \pi + \pi) = -\pi$ curvature. Figure 48 (left) shows the two points A and B where the vertices lie, the dotted lines representing the two edges, and the continuous line representing the two lines, compressed and overlapped. Figure 48 (right) also shows the longer edge pinched, and the intersection of three points $p_1, p_2, p3$ with $\kappa(p_1) = \pi, \kappa(p_2) = \pi, \kappa(p_3) = \pi$
Case $k > 0$. We assume that the lemma holds for $k - 1$, i.e. there is no Alexandrov gluing for a polygon with two concave vertices and $k - 1$ convex vertices. Say we zip at $v1$ (analogously for $v2$). If $v2$ lies at the end of one of edges adjacent in $v1$, then it is not possible to glue. If $v2$ lies somewhere else on the boundary, then the zip will result the result is a concave vertex $v2$, in a polygon with $k - 1$. This is our induction hypothesis, thus this state does not result in an Alexandrov gluing. The only option left is to glue one or more convex vertices into $v1$, then zip the difference. Gluing $n > 0$ vertices will result in n "holes" in the current shape, each hole being a sub-polygon of the initial polygon. In all of these polygons, there now exists a concave vertex obtained from the gluing. Lastly, one of these sub-polygons contains $v2$, therefore all of these polygons have two concave vertices and $k - m, m > 0$ convex vertices. We have now reached the induction hypothesis again. □

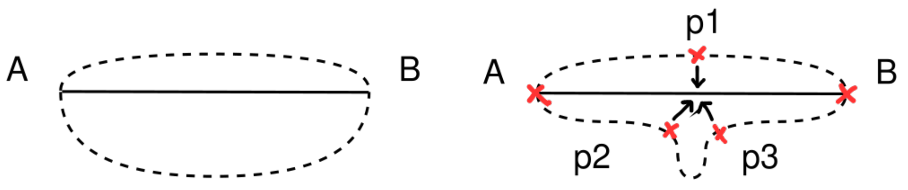


Figure 48: Case k=0

In conclusion, we can fold all convex and some non-convex polygons using Perimeter halving, and all matching edge-lengths polygons using a dynamic programming algorithm. Most randomly generated polygons don't fold into a convex polyhedron.

Chapter 5

Practical Applications of Folding

As we have seen in the previous two chapters, folding and unfolding have been studied extensively in a theoretical setting. In this chapter of the thesis, we will look at practical applications, without going into the mathematics or physics behind them.

This chapter is structured in various sections, each looking at a specific use case of folding/unfolding. An overview is provided without an extensive discussion or reasoning; these fall outside of the scope of computer science and of this thesis.

5.1 Linkages

A linkage is made of rigid bars (segments) connected in a joint (vertex), allowing the bars to rotate around the joint, while the bar itself doesn't bend. Figure 49 shows a linkage; the x-marks are stationary, while the dots change their position. The dotted line represents the movement of the right-most joint.

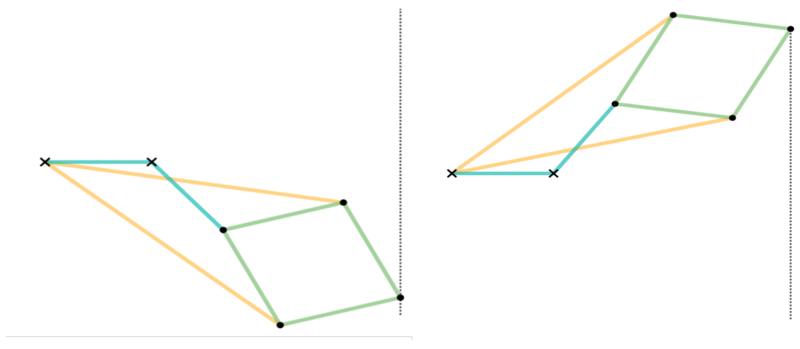


Figure 49: Two states in the movement of the linkage

This linkage is known as the "Peaucellier–Lipkin" linkage and it was invented in 1864. It could convert rotary motion into a perfect straight-line motion, and it was revolutionized steam engines at the time, which were previously inefficient at the motion conversion [21].

Linkages have been studied and used in practice for many years. We will now look at some of their applications, but for more details we recommend the chapter dedicated on linkages alone from O'Rouke and Demaine's work [13], but also the most extensive work we could find on linkages, which could as well serve as material for an undergraduate physics or mathematics course [27].

5.1.1 Robotics

Most robot mechanisms use linkages (e.g. excavator arm) that are based on linkage folding. This concept was first introduced in late 80's [26], and has been proven since to be effective (an interesting PhD thesis analyzes three folding methods of these king of robotic arms [6]).

Figure 50: Robot arm, adapted from [13], page 12

5.1.2 Writing Tool

An interesting historical machinery is an polygraph, which consists of a linkage-based tool used to copy text. It was first invented by John Isaac Hawkins, and it had two or more pens connected by linkages, mimicking the same movement, based the the movement made by the writer, holding one pen [28].

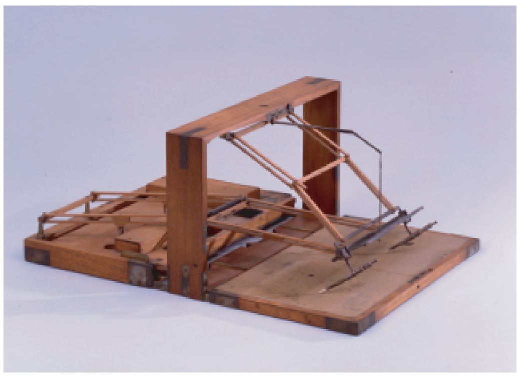


Figure 51: Polygraph, adapted from [13], page 13

5.1.3 Biology

While a computer scientist might not immediately relate to the term, a biologist often associates "folding" with protein folding. Similarly, when describing a linkage, a biologist is likely to also think of chemical molecules or DNA structures.

Proteins are, from a structural point of view, linkages. The protein folding problem is the equivalent of the convex polyhedra unfolding problem, in the sense that various questions have been posed by scientists over the last decades (e.g. speed of folding, end result of folding), yet no answers have been found [15]: it is not yet possible to predict what a protein will fold into.

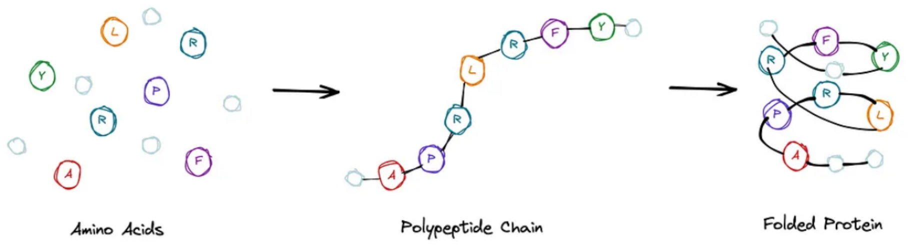


Figure 52: Protein folding[1]

Another use of folding is in DNA analysis, where it allowed scientists to create precise molecular structures that can be tailored with extreme precision down to a few nanometers [36].

[1]https://medium.com/qiskit/qiskits-protein-folding-module-has-moved-here-s-how-to-use-it-991b32381933

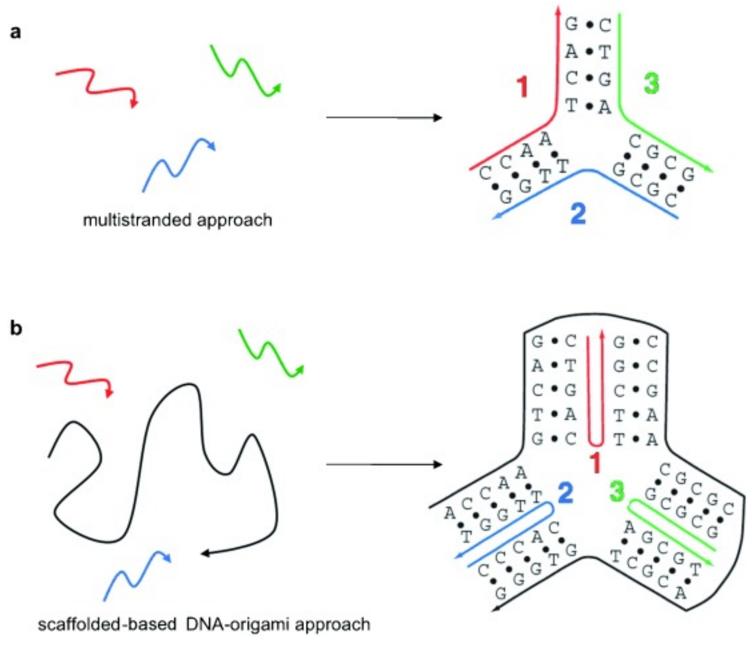


Figure 53: DNA folding

5.2 Medicine

Origami-based principles (paper folding) have been directly applied to create biomedical devices, particularly at very small scales, in orders of nanometer (1 millimeter = 1 000 000 nanometers). Two fascinating papers on the topic have been written by Ahmed et al. [1] and Johnson et al. [22].

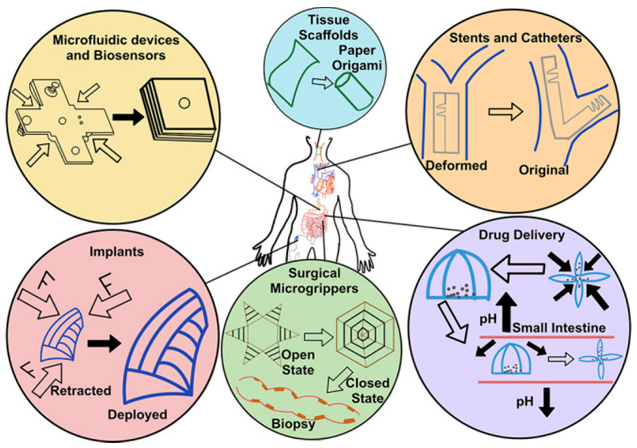


Figure 54: Various biomedical devices based on origami [1]

5.3 Manufacturing

In manufacturing, folding techniques are crucial for creating complex structures from simple sheets, enhancing efficiency and reducing material waste. A 3D part is approximated as a polyhedron, its surface is mapped to a collection of 2D flat patterns, each is cut from a sheet of material and folded [17].

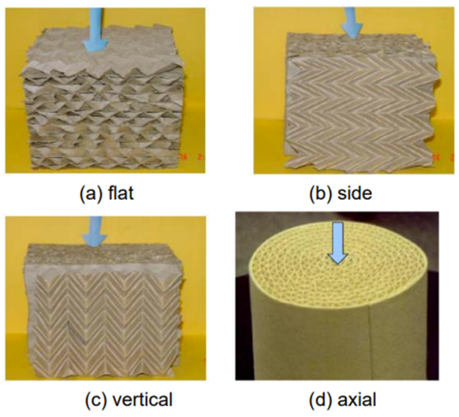


Figure 55: Chevron pattern, folded

In sheet-metal manufacturing, folding allows for the fabrication of parts for automotive and aerospace industries. This process is guided by algorithms that ensure optimal folds, minimizing the need for welding and fastening, thereby streamlining production lines and improving product integrity [4].

5.4 Airbag Design

Airbags are bags which are inflated within milliseconds with air to protect the passengers of a car in case of a crash. In its compact form, it needs to take as little space as possible. When the air is pushed into it, it should totally and efficiently unfold. Thus, scientists have turned to origami in order to find reliable folding methods [8].

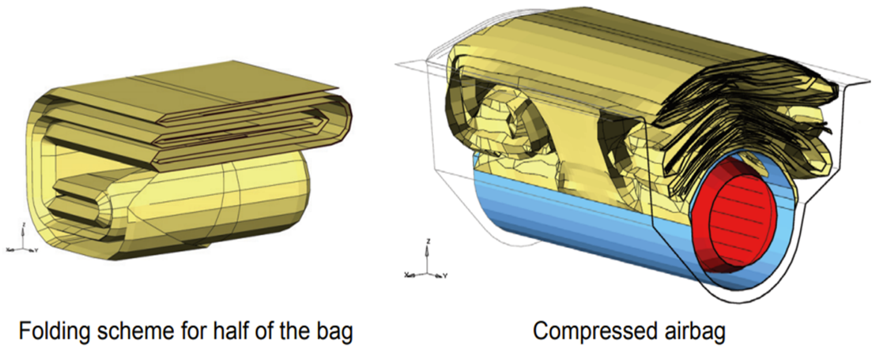


Figure 56: Airbag folding pattern [31]

5.5 Architectural Design

Applying the concept of folding in architecture is not straightforward, as buildings themselves do not fold. However, the idea of folding is utilized in various ways. For instance, it can be employed to enhance the visualization skills that architects need to effectively design building schematics. [18].

Another use case is artistic: building design. Origami has its roots in arts, and these beautiful shapes and figures have inspired architects to create buildings that are structurally stable but also pleasing to the eye, from the inside and the outside [40, 10].

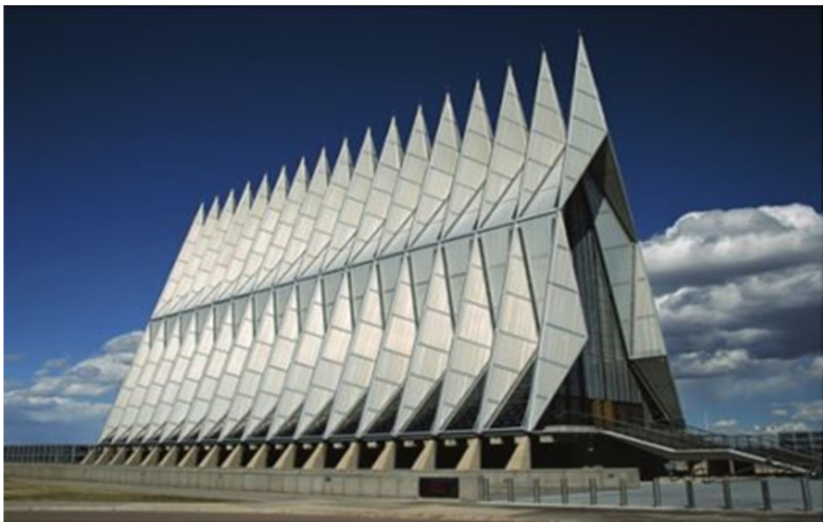


Figure 57: Air Force Academy Chapel, Colorado Springs, USA [38]

5.6 Spatial Engineering

In spatial engineering and spacecraft design, folding technologies play a pivotal role in optimizing the deployment of structures in the constrained environments of space missions [20]. These applications include the compact folding of solar panels and satellite antennae, which must be tightly packed for launch and then efficiently deployed once in orbit.

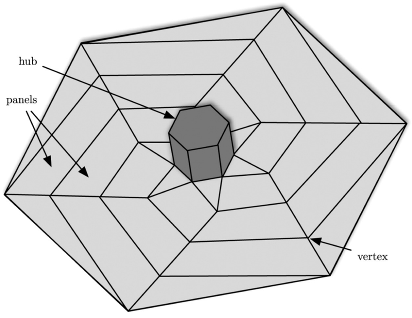


Figure 58: An satellite panel that can fold in a circular manner around its hub

5.7 Origami Art and Design

There have been entire books written on the topic of computational origami [44]. In this section, I will discuss *Origamizer*, an open-source software[2] that can generate folding patterns (similar to nets) for any shape, whether convex or non-convex [14]. Although the folding process for nets does not allow overlaps, paper has more relaxed folding rules compared to polygons and overlapping material (paper) is not a problem. A famous example used in the referenced paper [14] is the Stanford Bunny. The paper unfolding of the Stanford Bunny, as computed by the Origamizer, is shown in the image below.

[2]Note that the tilde might render to a different ascii character when copy-pasted: https://origami.c.u-tokyo.ac.jp/∼tachi/software/

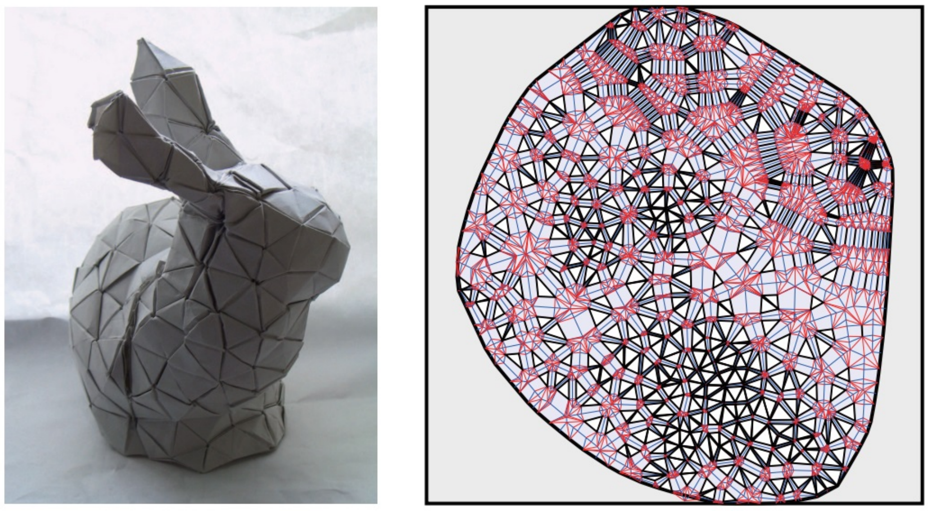


Figure 59: Origami pattern that results in a bunny

Chapter 6

Conclusions

This thesis focused on the sub-field of Computational Geometry known as Polyhedra Folding. The primary research question aimed to explore the methodologies for polygon folding and polyhedra unfolding, providing a bridge for novice researchers, particularly third-year Computer Science students, to understand this complex field.

The thesis provided the minimum necessary terminology in the *Background Knowledge* chapter, dissected core research into unfolding and folding in subsequent chapters, and informally described several practical applications of the theory.

However, this research faced several limitations. The scope was restricted to a review of existing literature without experimental validation of algorithms. Additionally, the complexity of some theoretical concepts could not be entirely simplified without losing depth.

Future research could build upon this foundation by experimentally validating the discussed algorithms, for example by implementing them in a programming language such as C++ and observing their behaviour, or writing a library that could simulate the methods. A novice researcher reading this thesis could also proceed to deepen their knowledge by analyzing the original sources.

In conclusion, this thesis provides a foundational resource that demystifies polygon and polyhedron folding, helping novice researchers navigate and understand the intricacies of this fascinating field. By building on this work, future research can continue to advance the practical and theoretical understanding of computational geometry.

Bibliography

[1] Abdor Rahman Ahmed, Olivia C Gauntlett, and Gulden Camci-Unal. Origami-inspired approaches for biomedical applications. *ACS omega*, 6(1):46–54, 2020.

[2] A. D. Aleksandrov and Lev Arkadevich Kaluzhnin. *Die innere Geometrie der konvexen Flächen.* January 1955.

[3] Alexandr D Alexandrov. *Convex polyhedra*, volume 109. Springer, 2005.

[4] Muhammad Ali Ablat and Ala Qattawi. Finite element analysis of origami-based sheet metal folding process. *Journal of Engineering Materials and Technology*, 140(3):031008, 2018.

[5] Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, SCG '91, New York, NY, USA, 1991. Association for Computing Machinery.

[6] Nadia M Benbernou. *Geometric algorithms for reconfigurable structures.* PhD thesis, Massachusetts Institute of Technology, 2011.

[7] Therese C. Biedl, Erik D. Demaine, Martin L. Demaine, Anna Lubiw, Mark H. Overmars, Joseph O'Rourke, Steve Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. *Canadian Conference on Computational Geometry*, January 1998.

[8] Christoffer Cromvik and Kenneth Eriksson. *Airbag folding based on origami mathematics.* Chalmers University of Technology Göteborg, Sweden, August 2006.

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

[10] Pierluigi D'Acunto and Juan José Castellón González. Folding augmented: A design method for structural folding in architecture. In *The 6th Interntaional Meeting on Origami in Science, Mathematics and Education: Program and Abstracts: held at The University of Tokyo, August 10-13, 2014.* 6OSME Program and Organizing Committees, 2014.

[11] Nihad E Daidzic. Long and short-range air navigation on spherical earth. *International Journal of Aviation Aeronautics and Aerospace*, 4(1), 2017.

[12] Mirela Damian and Henk Meijer. Edge-unfolding polycubes with orthogonally convex layers. *arXiv preprint arXiv:2407.01326*, 2024.

[13] Erik D Demaine and Joseph O'Rourke. *Geometric folding algorithms: linkages, origami, polyhedra.* Cambridge university press, 2007.

[14] Erik D. Demaine and Tomohiro Tachi. Origamizer: a practical algorithm for folding any polyhedron. *Symposium on Computational Geometry*, page 16, January 2017.

[15] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

[16] Albrecht Dürer. *Underweysung der messung, mit den zirckel un richtscheyt, in Linien ebnen unnd gantzen corporen.* 1525.

[17] EA Elsayed and Basily B Basily. Applications of folding flat sheets of materials into 3-d intricate engineering designs. *Department of Industrial and Systems Engineering Rutgers University*, 2007.

[18] Carmen Escoda Pastor. Origami as a tool for three-dimensional architectonic thought. In *Graphic Imprints*, pages 1554–1565, Cham, 2019. Springer International Publishing.

[19] P. J. Federico. *Descartes on Polyhedra: A study of the de solidorum elementis.* Springer, Berlin, December 1982.

[20] JoAnna Fulton and Hanspeter Schaub. Forward dynamics analysis of origami-folded deployable spacecraft structures. *Acta Astronautica*, 186:549–561, 2021.

[21] R. S. Hartenburg, J. Denavit, and F. Freudenstein. Kinematic synthesis of linkages. *Journal of Applied Mechanics*, 32(2):477, June 1965.

[22] Meredith Johnson, Yue Chen, Sierra Hovet, Sheng Xu, Bradford Wood, Hongliang Ren, Junichi Tokuda, and Zion Tsz Ho Tse. Fabricating biomedical origami: a state-of-the-art review. *International journal of computer assisted radiology and surgery*, 12:2023–2032, 2017.

[23] Shoshichi Kobayashi. Invariant distances on complex manifolds and holomorphic mappings. *Journal of the Mathematical Society of Japan*, 19(4):460–480, 1967.

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

[25] Brendan Lucier. *Unfolding and reconstructing Polyhedra.* PhD thesis, University of Waterloo, January 2006.

[26] Vladimir Lumelsky and Kang Sun. A unified methodology for motion planning with uncertainty for 2d and 3d two-link robot arm manipulators. *The International journal of robotics research*, 9(5):89–104, 1990.

[27] J Michael McCarthy and Gim Song Soh. *Geometric design of linkages*, volume 11. Springer Science & Business Media, 2010.

[28] R. M. S. McDonald. The papers of thomas jefferson, retirement series. *the Journal of American History*, 100(1):190–191, June 2013.

[29] Koichi Mizunashi, Takashi Horiyama, and Ryuhei Uehara. Efficient algorithm for box folding. *Journal of Graph Algorithms and Applications*, 24(2):89–103, January 2020.

[30] David Mount. Storing the subdivision of a polyhedral surface. In *Proceedings of the second annual symposium on Computational geometry*, pages 150–158, 1986.

[31] Krystoffer Mroz and Bengt Pipkorn. Mathematical modelling of the early phase deployment of a passenger airbag–folding using origami theory and inflation using ls-dyna particle method. In *Sixth European LS-DYNA users conference*, 2007.

[32] Joseph O'Rourke, Subhash Suri, and Heather Booth. Shortest paths on polyhedral surfaces. In *STACS 85: 2nd Annual Symposium on Theoretical Aspects of Computer Science Saarbrücken, January 3–5, 1985 2*, pages 243–254. Springer, 1985.

[33] Joseph O'Rourke and Costin Vilcu. Skeletal cut loci on convex polyhedra. *arXiv preprint arXiv:2312.01534*, 2023.

[34] Val Pinciu. On the fewest nets problem for convex polyhedra. *Canadian Conference on Computational Geometry*, page 21–24, January 2007.

[35] Sebastian Pütz. Navigation control and path planning for autonomous mobile robots: Dissertation abstract. *KI-Künstliche Intelligenz*, pages 1–4, 2024.

[36] Barbara Saccà and Christof M Niemeyer. Dna origami: the art of folding dna. *Angewandte Chemie International Edition*, 51(1):58–66, 2012.

[37] Wolfram Schlickenrieder. Nets of polyhedra. *Master's Thesis, Technische Universität Berlin*, 1997.

[38] N. Šekularac, J. Ivanovic-Šekularac, and J. Cikic-Tovarovic. Folded structures in modern architecture. *Facta universitatis-Series: Architecture and Civil Engineering*, 10:1–16, 2012.

[39] Micha Sharir and Amir Schorr. On shortest paths in polyhedral spaces. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 144–153. SIAM Journal on Computing, 1984.

[40] Tao Shen and Yukari Nagai. An overview of folding techniques in architecture design. *World Journal of Engineering and Technology*, 5(3):12–19, 2017.

[41] Geoffrey C Shephard. Convex polytopes with convex nets. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 78, pages 389–403. Cambridge University Press, 1975.

[42] Takumi Shiota and Toshiki Saitoh. Overlapping edge unfoldings for convex regular-faced polyhedra. *Theoretical Computer Science*, 2024.

[43] Ernst Steinitz. *Vorlesungen über die Theorie der Polyeder: unter Einschluß der Elemente der Topologie*, volume 41. Springer-Verlag, 2013.

[44] Ryuhei Uehara. *Introduction to Computational Origami.* Springer, 2020.

[45] Lars Zawallich. Unfolding polyhedra via tabu search. *The Visual Computer*, 2024.