Convexifying Polygons Without Losing Visibilities

Oswin Aichholzer* Greg Aloupis† Erik D. Demaine‡ Martin L. Demaine‡ Vida Dujmović§
Ferran Hurtado¶ Anna Lubiw‖ Günter Rote** André Schulz†† Diane L. Souvaine‡‡
Andrew Winslow‡‡

Abstract

We show that any simple n-vertex polygon can be made convex, without losing internal visibilities between vertices, using n moves. Each move translates a vertex of the current polygon along an edge to a neighbouring vertex. In general, a vertex of the current polygon represents a set of vertices of the original polygon that have become co-incident.

We also show how to modify the method so that vertices become very close but not co-incident, in which case we need $O(n^2)$ moves, where each move translates a single vertex.

The proof involves a new visibility property of polygons, namely that every simple polygon has a *visibility-increasing edge* where, as a point travels from one endpoint of the edge to the other, the visibility region of the point increases.

1 Introduction

There are many interesting problems about reconfiguring geometric structures while maintaining some properties. Examples include: flips in triangulations [5], pushing and sliding block puzzles [16], morphing of polygons and planar graphs [18, 21], and linkage reconfiguration [7, 23]. Reconfiguration has also been studied outside the geometric domain [19].

This paper is about *convexifying* a simple polygon, i.e., continuously transforming the polygon to a convex polygon while maintaining simplicity. If no other structure must be maintained, this can be done in a trivial way, moving only one vertex at a time. When edge lengths must be maintained, this is a major result, namely the Carpenter's Rule Theorem [7, 23], and the reconfiguration process involves moving all vertices simultaneously.

In the Open Problem session at CCCG 2008 [11], Satyan Devadoss asked whether a polygon can be convexified without losing internal visibility between any pair of vertices, and in particular, whether this can be done by moving only one vertex at a time [12]. We give a positive answer. We first consider a version of the problem where vertices may become co-incident during the transformation, so one vertex of the polygon in general represents a set of vertices of the original polygon. We show that any polygon can be convexified by a sequence of n moves, where each move strictly increases the set of pairs of vertices that are internally visible, and each move translates one vertex along an edge of the polygon to a neighbour. In terms of the original polygon, each move translates a set of vertices along a straight line to join another set of vertices.

In Section 3 we modify our method so that vertices become very close but not coincident. In this case, we need $O(n^2)$ moves, each moving one vertex. Internal vertex visibilities are never lost, but a single move does not necessarily add any internal vertex visibilities.

Our main tool, which may be of independent interest, is to show that every polygon has a *visibility-increasing edge* where, as a point travels from one endpoint of the edge to the other, the visibility region of the point increases.

Previous Work

In the original model where coincident vertices are not allowed, Aichholzer et al. [2] showed that any monotone polygon can be convexified without losing vertex visibilities. Their transformation moves one vertex at a time, but the number of vertex moves is not polynomially bounded. If all vertices may move simultaneously,

*Institute for Software Technology, University of Technology Graz, Austria, oaich@ist.tugraz.at. Partially supported by the FWF under grant S9205-N12 NRN Industrial Geoemtry.

†Département d'Informatique, Université Libre de Bruxelles, aloupis.greg@gmail.com

‡MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, {edemaine, mdemaine}@mit.edu

§School of Computer Science, Carleton University, vida@scs.carleton.ca

¶Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, ferran.hurtado@upc.edu. Partially supported by projects MICINN MTM2009-07242 and Gen. Cat. DGR 2009SGR1040

‖School of Computer Science, University of Waterloo, alubiw@uwaterloo.ca

**Institut für Informatik, Freie Universität Berlin, Takustraße 9, 14195 Berlin, Germany. rote@inf.fu-berlin.de

††Institut für Mathematische Logik und Grundlagenforschung, Universität Münster, Germany, andre.schulz@uni-muenster.de

‡‡Department of Computer Science, Tufts University, {dls, awinslow}@cs.tufts.edu. Research supported in part by NSF grants CCF-0830734 and CBET-0941538.

they observe that a monotone polygon can be convexified in one move. They also show that, even for monotone polygons, it is not always possible to move just one vertex and strictly increase the set of vertex visibilities. Note that such an example depends crucially on prohibiting coincident vertices! If vertices are allowed to be coincident, our result shows that for any simple polygon, it is possible to move one vertex until it gains a new neighbour in the visibility graph.

The issue of allowing/disallowing coincident vertices has arisen before in problems of transforming (or "morphing") polygons and straight-line graph drawings. Cairns [6] showed how to transform between any two straight-line planar triangulations that are combinatorially the same, using a sequence of moves each of which translates one vertex onto another (or the reverse). He then comments that it is possible to avoid coincident vertices by keeping them a small distance apart. A somewhat similar issue comes up in the result of Guibas and Hershberger [17] who show that for any two simple polygons on vertices $1, 2, \ldots, n$ such that edge $(i, i+1)$ has the same direction vector in both polygons, there is a morph between the polygons that preserves simplicity and the direction vectors of edges. Their method moves vertices infinitesimally close together and operates on the infinitesimal structures.

The question of moving only one vertex at a time has recently been settled in independent work by Ábrego et al. [1], who show that if a there is a transformation that convexifies a polygon without losing vertex visibilities, then the transformation can be accomplished by moving only one vertex at a time.

Although not directly relevant to this paper, we note that there is a considerable body of work on making polygons convex by means of "pivot" operations, such as *flips* [13, 8, 15, 25] and *flipturns* [3, 4].

For background on visibility graphs of polygons, see the books by Ghosh [14] and O'Rourke [22].

Definitions

Two points inside a polygon P are *visible* if the line segment between them is contained in the closed polygon. Given this definition, we will now use "visibility" rather than "internal visibility". We will assume that the input polygon does not have three or more collinear vertices. It is possible to perturb a polygon to achieve this without losing internal vertex visibilities. Note the consequence that if two vertices are visible, then the line segment between them does not go through another vertex. For point p in P, the *visibility region* of p, denoted $V(p)$, is the set of points in P visible from p.

Let a be a reflex vertex with neighbours b and b' on the polygon boundary. Extend a line segment from b to a and beyond, until it first hits the polygon boundary at p. Define $\mathrm{Pocket}(b, a)$ to be the region bounded by the chain along the polygon boundary from a to p going through b', together with the line segment pa. We consider points along the line segment pa to be outside the pocket (i.e., the pocket is open along its "mouth"). In particular, a is outside $\mathrm{Pocket}(b, a)$. See the shaded region in Figure 1(a).

2 Convexifying polygons

Theorem 1 *An n-vertex polygon can be convexified in n moves, where each move strictly increases the set of pairs of visible vertices, and each move translates one vertex of the current polygon along an incident edge to a neighbour on the polygon boundary.*

The main tool in proving the theorem is the following. We prove that if a polygon is not convex then it has an edge along which visibility increases. More precisely, define an edge (u, v) to be a *visibility-increasing edge* if for every point p along the edge (u, v) we have $V(u) \subseteq V(p) \subseteq V(v)$, and there is a vertex in $V(v) - V(u)$.

We will use a stronger induction hypothesis to prove that every non-convex polygon has a visibility-increasing edge (u, v) where v is a reflex vertex. Note that the fact that v is reflex implies that there is a vertex in $V(v) - V(u)$.

Lemma 2 *Let P be a simple polygon with reflex vertex a and edge (b, a). Then there is a visibility-increasing edge (u, v) with v reflex and u, v exterior to $\mathrm{Pocket}(b, a)$ such that u does not see into $\mathrm{Pocket}(b, a)$.*

Proof. We prove the result by induction on the number of reflex vertices of the polygon exterior to the pocket. If (b, a) is a visibility-increasing edge, then it satisfies the lemma, since b does not see into $\mathrm{Pocket}(b, a)$. See Figure 1(a). This takes care of the base case where every vertex $v \neq a$ exterior to the pocket is convex.

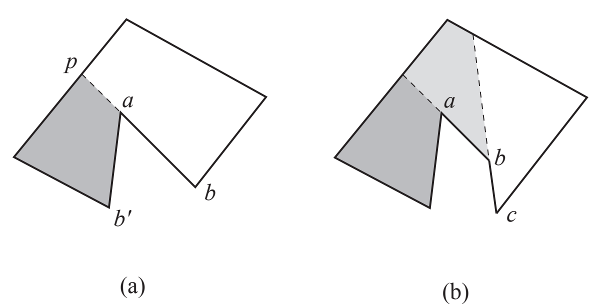

Figure 1: Visibility-increasing edges: (a) the edge (b, a) is a visibility-increasing edge; (b) vertex b is reflex, so we apply induction on (c, b).

If b is a reflex vertex then let c be the other neighbour of b (i.e., the neighbour not equal to a). See Figure 1(b). Then $\mathrm{Pocket}(c, b) \supseteq \mathrm{Pocket}(b, a)$. Also, note that the reflex vertex a is exterior to $\mathrm{Pocket}(b, a)$ and not exterior to Pocket(c, b). Therefore we can apply induction to conclude that there is a visibility-increasing edge (u, v) exterior to Pocket(c, b) such that v is reflex and u does not see into Pocket(c, b). Then u cannot see into Pocket(b, a), so (u, v) satisfies the lemma.

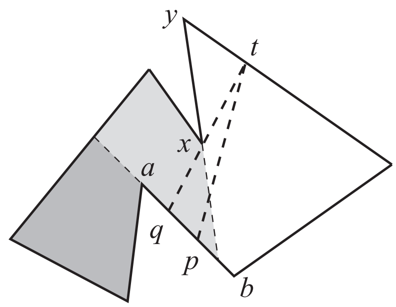

Figure 2: Visibility-increasing edges in the general case, where we apply induction on (y, x).

We are left with the case where b is a convex vertex but (b, a) is not a visibility-increasing edge. Note that because a is a reflex vertex, $V(a)$ contains a vertex not in $V(b)$. Therefore, the only way that (b, a) can fail to be visibility-increasing is that there is a point p on (b, a) and a point t on the boundary of P such that t sees p, but t does not see a. See Figure 2. Now we rotate the line through t and p about t until it hits the polygon boundary. More precisely, consider the first point q along the line segment pa such that the line segment qt does not lie in the interior of P. Then some vertex x lies on the line segment qt. Note that x must be a reflex vertex. There are two paths on the polygon boundary from x to t. Take the path that does not contain a, and let y be the neighbour of x on this path. (It may happen that $y = t$.) We will apply induction on the edge (y, x). Observe that Pocket$(y, x) \supseteq$ Pocket(b, a). Also, note that the reflex vertex a is exterior to Pocket(b, a) and not exterior to Pocket(y, x). Therefore we can apply induction to conclude that there is a visibility-increasing edge (u, v) exterior to Pocket(y, x) such that v is reflex and u does not see into Pocket(y, x). Then u cannot see into Pocket(b, a), so (u, v) satisfies the lemma. $\square$

Proof. [of Theorem 1] The proof is by induction on the number of vertices. If the polygon has three vertices then it is already convex. For the general case, if the polygon is convex then there is nothing to prove, so suppose there is a reflex vertex. Then by Lemma 2, there is a visibility-increasing edge (u, v). The plan is to move vertex u to vertex v, resulting in a simple polygon with fewer vertices on which we apply induction. See Figure 5. Let w be the other neighbour of u on the polygon boundary. We have $V(u) \subseteq V(v)$ and $w \in V(u)$, so w must be visible to v. In particular, u is a convex vertex and the line segment wv does not intersect the polygon boundary except at its endpoints. Therefore moving u to v results in a simple polygon. Observe that no vertex visibilities are affected by the move, except that u gains visibilities once it reaches v (if not before). Note that u may become collinear with two other vertices of the polygon at an intermediate point of the move, but this causes no problems. $\square$

3 Avoiding coincident vertices

In the previous section we showed how to convexify any polygon without losing internal visibilities, provided that vertices are allowed to become coincident. In this section we show how to avoid coincident vertices. Each set of coincident vertices from the previous method is replaced by a cluster of vertices that are close together but not coincident. One move of the previous method becomes $O(n)$ moves, each moving a single vertex. The total number of moves is therefore $O(n^2)$. Vertex visibilities are never lost, but a single move might not increase vertex visibilities.

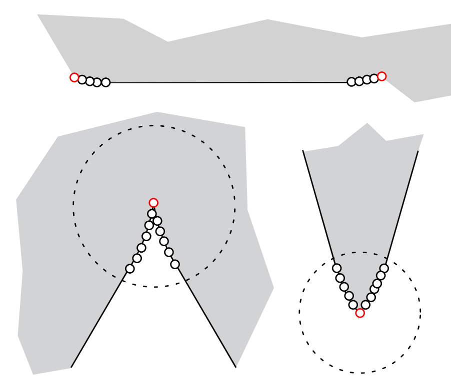

Figure 3: Cluster vertices along a single edge (top); a reflex cluster (left); and a convex cluster (right). Shaded areas indicate the interior of the polygon.

The basic idea is to replace an edge uv by a slightly outward-bent convex chain, with some points on a shallow convex curve close to u, and other points on a shallow convex curve close to v, see Figure 3 (top). In general, a *cluster* will consist of a *representative vertex* v, together with the vertices that have been moved to join v, and now lie on two convex curves incident to v. The representative vertex v will be at the same point in the plane as it was in the original polygon. If C is a cluster with representative vertex v, we will say that C is *the cluster of* v. Figure 3 depicts a *reflex* and a *convex* cluster. In a convex cluster all vertices see each other; in a reflex cluster only vertices in the same arc see each other, and the representative vertex sees the whole cluster.

All vertices of a cluster lie in the ε-neighbourhood of the representative vertex for some sufficiently small ε. In addition, all vertices of a cluster lie within some angle α of the original edge. See Figure 4(a).

We define values for ϵ and α that will work throughout the algorithm. As the convexification proceeds, edges between representative vertices of the intermediate polygons are always chords of the original polygon. We will take all the chords into account when we define ε and α. We choose ε small enough that visibility between two points in the ε-neighbourhoods of two vertices x and y behaves like visibility between x and y. Thus ε should be smaller than the distance between any vertex and a (non-coincident) chord or edge extension—see Figure 4(b). We choose α small enough that a representative vertex x does not block visibilities of vertices in its cluster, and that a convex vertex remains convex—see Figure 4(c). Apart from the constraints imposed by ϵ and α we are free to place the cluster vertices on any convex chain, and we will have occasion to alter the chain.

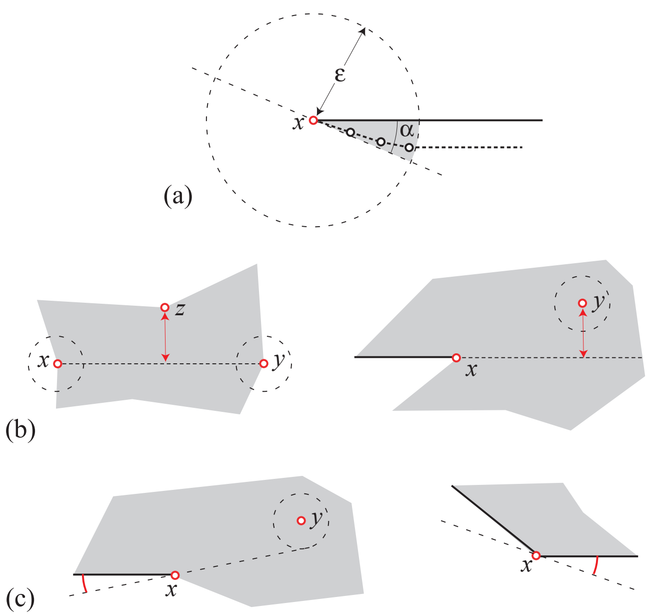

Figure 4: (a) Cluster vertices are located in the shaded region determined by ε and α; (b) Constraints on ε, which must be small enough that visibility from a point within an ϵ-neighbourhood of a vertex acts like visibility from the vertex; (c) Constraints on α, which must be small enough that a vertex x does not block visibility to its cluster.

We now consider the move operation from the previous section as it operates on clusters. The move operation always moves a convex vertex u to join a reflex vertex v. See Figure 5. The only other vertex affected by the move is w, the other neighbour of u, which forms a triangle with u and v. Suppose without loss of generality that v, u, w are in clockwise order around the polygon. When vertices are replaced by clusters, the vertices affected by the move are: all of u's cluster; the left part of v's cluster; and the right part of w's cluster. See Figure 6. Note that, although the original move always increased the set of vertices visible from u, the modified move will not necessarily increase visibility from u or any of its cluster, since we do not move any vertex all the way to v.

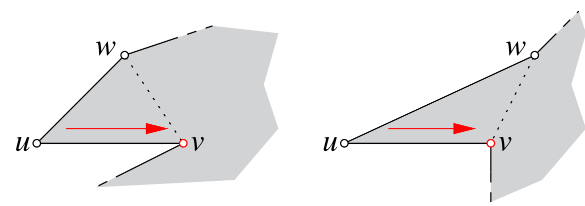

Figure 5: Moving vertex u along the visibility-increasing edge (u, v) affects vertices u, v, and w, which form a triangle. Vertex v may remain reflex (left) or become convex (right).

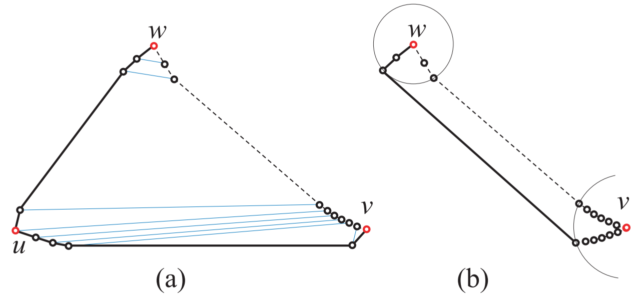

Figure 6: The operation from Figure 5 in the presence of cluster vertices: (a) the initial configuration, the final configuration shown with dashed lines, and the vertex correspondence shown with thin lines; (b) the intermediate configuration after moving u and its cluster close to v.

We show that the transformation of clusters as shown in Figure 6 can be accomplished by moving one vertex at a time. The first phase is to move u and its cluster close to v, in a configuration congruent to their final configuration. Move the vertices one by one starting with the vertex closest to v along the chain. Note that u loses its status as a representative vertex. The result of the first phase is shown in Figure 6(b). Note that the union of the initial and final positions of all the vertices that are moved in the first phase is in convex position. Therefore, convexity of the cluster and visibility within the cluster are maintained. Globally, as each cluster vertex moves from the neighbourhood of u's initial position to v's neighbourhood, its visibility changes exactly as u's visibility changed in the original non-cluster move (stopped just before reaching v).

In the second phase (from Figure 6(b) to the final configuration) the transformation we wish to realize is a counterclockwise rotation of w's right cluster and a clockwise rotation of v's left cluster to their final positions. We describe how to do this for v's left cluster. In the first step, move the vertices of v's left cluster (one by one) close enough to v that their new positions and their final positions are in convex position, as shown in Figure 7. In the second step, move the vertices one by one to their final positions, starting with the vertex farthest from v along the chain. Convexity of the cluster (and hence visibility within the cluster) is maintained during the second step because the union of the initial and final positions of all moved vertices is in convex position. Global visibilities may be gained but are never lost.

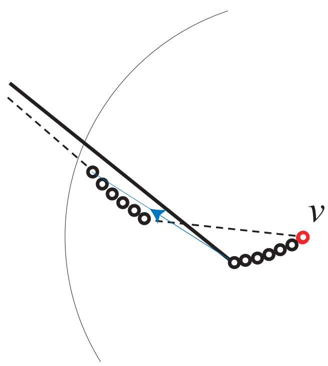

Figure 7: Adjusting the position of v's left cluster vertices. All movement takes place within the ϵ-neighbourhood of v. The first vertex move is shown with a thin directed line. Note that this figure is not to scale since the angle α should be much smaller.

From the above ideas, we obtain the following result.

Theorem 3 *An n-vertex polygon can be convexified in $O(n^2)$ moves, so that visibilities between vertices are never lost, and vertices never become coincident. Each move is a translation of a single vertex.*

4 Discussion and Open Problems

We have shown that any simple n-vertex polygon can be convexified in $O(n^2)$ single-vertex moves without ever decreasing the visibility graph, answering a question posed by Devadoss et al. [12]. If coincident vertices are allowed, then n moves suffice, and each move strictly increases the visibility graph.

In the same paper, Devados et al. ask about transforming a polygon to decrease the visibility graph: can any simple polygon be transformed to a polygon whose visibility graph is a triangulation without ever increasing the visibility graph? This question remains open.

For orthogonal polygons, it would be desirable to maintain orthogonality. We conjecture than every simple orthogonal polygon can be convexified (i.e., transformed to a rectangle) without losing visibilities, while maintaining orthogonality. A minimal motion that maintains orthogonality is to move one edge orthogonal to itself (i.e., a horizontal edge moves vertically, and vice versa). However, Figure 8 shows an example where no edge can be moved orthogonally to gain visibilities.

It is possible that the current result can be generalized to straight line drawings of planar graphs: Given a planar graph embedded in the plane as a straight-line drawing, is it possible to transform the drawing so that every internal face becomes convex, while remaining straight-line planar, and without losing internal visibilities? Our result is the special case where the drawing has only one internal face. The fact that such a transformation is possible, ignoring visibility constraints, is not at all obvious, but follows from the result by Thomassen [24], who showed (based on a result of Cairns [6]) that there is a transformation between any two straight-line planar drawings of the same embedded graph that preserves straight-line planarity. Vertices become coincident during this transformation, although that can be avoided by keeping them close but distinct. The number of vertex movements is not polynomially bounded. For further discussion on morphing of graph drawings, see [20, 21].

Finally, we make two remarks about our result on the existence of a visibility-increasing edge in any simple polygon. Since good things (like ears of polygons) come in pairs, it is natural to ask whether every simple polygon has *two* visibility-increasing edges.

Visibility-increasing edges may have other uses in the study of visibility graphs. A major open question is whether visibility graphs of polygons can be recognized in polynomial time (with or without the information about which edges form the polygon boundary). This is Problem 17 in the Open Problems Project [9].

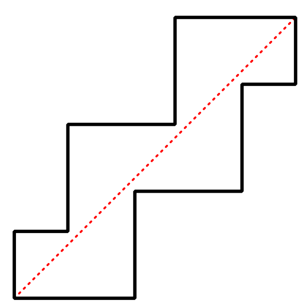

Figure 8: An orthogonal polygon where no single edge can be moved orthogonally to gain visibilities.

5 Acknowledgments

This work was begun at the 26th Bellairs Workshop on Computational Geometry, co-organized by Erik Demaine and Godfried Toussaint. We thank the other participants of the workshop for stimulating discussions.

References

[1] B. M. Ábrego, M. Cetina, J. Leaños, and G. Salazar. Visibility-preserving convexifications using single-vertex moves. `http://arxiv.org/pdf/1105.3435v1`.

[2] O. Aichholzer, M. Cetina, R. Fabila-Monroy, J. Leanos, G. Salazar, and J. Urrutia. Convexifying monotone polygons maintaining internal visibility. Extended abstract, XIV Spanish Meeting on Computational Geometry, Alcalá de Henares, Spain, pages 35–38, 2011.

[3] O. Aichholzer, C. Cortés, E. D. Demaine, V. Dujmovic, J. Erickson, H. Meijer, M. Overmars, B. Palop, S. Ramaswami, and G. Toussaint. Flipturning polygons. *Discrete & Computational Geometry*, 28:231–253, 2002.

[4] T. Biedl. Polygons Needing Many Flipturns. *Discrete & Computational Geometry*, 35:131–141, 2006.

[5] P. Bose and F. Hurtado. Flips in planar graphs. *Computational Geometry*, 42:60–80, 2009.

[6] S. S. Cairns. Deformations of plane rectilinear complexes. *Amer. Math. Monthly*, 51(5):247–252, 1944.

[7] R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete & Computational Geometry*, 30:205–239, 2003.

[8] B. de Sz.-Nagy. Solution of problem 3763. *Amer. Math. Monthly*, 49:176–177, 1939.

[9] E. D. Demaine, J. Mitchell, and J. O'Rourke. The open problems project. `http://maven.smith.edu/~/TOPP/`, May 2010.

[10] E. D. Demaine and J. O'Rourke. Geometric Folding Algorithms: Linkages, Origami, Polyhedra. Cambridge University Press, 2007.

[11] E. D. Demaine and J. O'Rourke. Open problems from CCCG 2008. In *Proceedings of the 21st Canadian Conference on Computational Geometry (CCCG)*, pages 75–78, 2009.

[12] S. L. Devadoss, R. Shah, X. Shao, and E. Winston. Visibility graphs and deformations of associahedra. `http://arxiv.org/abs/0903.2848`.

[13] P. Erdős. Problem 3763. *Amer. Math. Monthly*, 42:42, 1935.

[14] S. Ghosh. *Visibility Algorithms in the Plane.* Cambridge University Press, 2007.

[15] B. Grünbaum. How to convexify a polygon. *Geocombinatorics*, 5:24–30, 1995.

[16] R. A. Hearn and E. D. Demaine *Games, puzzles, and computation.* A. K. Peters, Ltd., 2009.

[17] L. Guibas and J. Hershberger. Morphing simple polygons. In *Proceedings of the 10th Annual Symposium on Computational Geometry*, SCG '94, pages 267–276, New York, NY, USA, 1994. ACM.

[18] H. N. Iben, J. F. O'Brien and E. D. Demaine. Refolding Planar Polygons. In *Proceedings of the 22nd Annual Symposium on Computational Geometry*, pages 71–79, 2006. ACM.

[19] T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara and Y. Uno On the Complexity of Reconfiguration Problems. *Theoretical Computer Science*, in press.

[20] A. Lubiw and M. Petrick. Morphing planar graph drawings with bent edges. *Electronic Notes in Discrete Mathematics*, 31:45 – 48, 2008. The International Conference on Topological and Geometric Graph Theory.

[21] A. Lubiw, M. Petrick, and M. Spriggs. Morphing orthogonal planar graph drawings. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '06, pages 222–230, 2006. ACM.

[22] J. O'Rourke. *Art gallery theorems and algorithms.* Oxford University Press, Inc., 1987.

[23] I. Streinu. Pseudo-triangulations, rigidity and motion planning. *Discrete & Computational Geometry*, 34:587–635, 2005.

[24] C. Thomassen. Deformations of plane graphs. *Journal of Combinatorial Theory, Series B*, 34(3):244 – 257, 1983.

[25] G. Toussaint. The Erdös–Nagy theorem and its ramifications. In *Proceedings of the 11th Canadian Conference on Computational Geometry (CCCG)*, pages 219–236, 1999.