7-13-2010

On Flat Polyhedra Deriving From Alexandrov's Theorem

Joseph O'Rourke
Smith College, jorourke@smith.edu

On Flat Polyhedra deriving from Alexandrov's Theorem

Joseph O'Rourke*

September 10, 2015

Abstract

We show that there is a straightforward algorithm to determine if the polyhedron guaranteed to exist by Alexandrov's gluing theorem is a degenerate flat polyhedron, and to reconstruct it from the gluing instructions. The algorithm runs in $O(n^3)$ time for polygons whose gluings are specified by n labels.

1 Introduction

A theorem of Alexandrov says that any "gluing" of polygons that satisfies three conditions corresponds to a unique convex polyhedron. The theorem includes flat doubly covered convex polygons as among the possible "convex polyhedra" whose existence is guaranteed by the theorem. In this note we provide an algorithm that detects if a gluing will produce such a flat polyhedron, and if so, constructs it.

Alexandrov's 1941 theorem is described in his 1950 book *Convex Polyhedra*, recently translated into English [Ale05]. Descriptions may be found in [DO07, Sec. 23.3] and [Pak10, Sec. 37]. Here we give a brief statement of the theorem. Define an *Alexandrov gluing* of a collection of polygons one as satisfying these conditions:

1. The gluing matches all the perimeters of the polygons by identifying which points *glue* to which. A case of special interest is when there is just one polygon, whose perimeter is glued to itself. Isolated points may have no match, where the boundary "zips" closed in a neighborhood of those points.

2. The gluing creates no more than 2π surface angle surrounding any point of the resulting manifold.

3. The gluing results in manifold that is homeomorphic to a sphere.

*Department of Computer Science, Smith College, Northampton, MA 01063, USA. orourke@cs.smith.edu.

These three conditions are obviously necessary for the manifold to be a convex polyhedron. Alexandrov's Theorem says that these conditions are also sufficient:

Theorem 1 (Alexandrov) *Any Alexandrov gluing corresponds to a unique convex polyhedron (where a doubly covered polygon is considered a polyhedron).*

Alexandrov's proof is a difficult existence proof and gives little hint of the structure of the polyhedron guaranteed by the theorem. Recently, Bobenko and Izmestiev found an intricate but constructive proof of the theorem, which can be used to reconstruct the 3D polyhedron as the solution of a particular differential equation [BI08]. They have implemented an approximate numerical solution of this equation in publicly available software. See [O'R07] for a high-level description of their proof.

The flat polyhedra permitted by Alexandrov's Theorem are necessary. For example, folding a square across a diagonal constitutes an Alexandrov gluing, and results in a flat doubly covered isosceles right triangle. The purpose of this note is to isolate the degenerate flat-polyhedron case of Alexandrov's Theorem, and show that detection and reconstruction are possible by a straightforward algorithm that need not confront the complexities of the full theorem.

2 What is n?

Before describing the algorithm, we first address a confusing issue[1] concerning the appropriate value of n for this question, the primary combinatorial count for expressing complexity. There are four possible n's:

1. n_p: the total number of vertices in the collection of polygons.
2. n_g: the number of gluing labels defining the gluing instructions.
3. n_s: the number of vertices on the surface of the polyhedron $\mathcal{P}$.
4. n_c: the number of corners or *cone points* on the surface of $\mathcal{P}$.

A cone point is a point surrounded by strictly less than 2π of surface. Although we are only interested in order of magnitudes when claiming a time complexity of $O(n^3)$, it is not the case that all these possible n's are necessarily linearly related.

For n_p, it is natural to count only vertices whose internal angle differs from π. But n_g could be arbitrarily larger than n_p. An example is shown in Figure 1 (other examples may be found in [ADD+11]). Here a rectangle (n_p=4) is wrapped in a spiral to form a doubly covered trapezoid (n_c=4). But n_g=12 segments around the boundary of the rectangle are labeled to specify the Alexandrov gluing, and it is clear that more spiraling could raise n_g arbitrarily. The gluing reduces the points on $\mathcal{P}$ delimiting the "faces" by possibly half, in this

[1] First brought to my attention by Anna Lubiw.

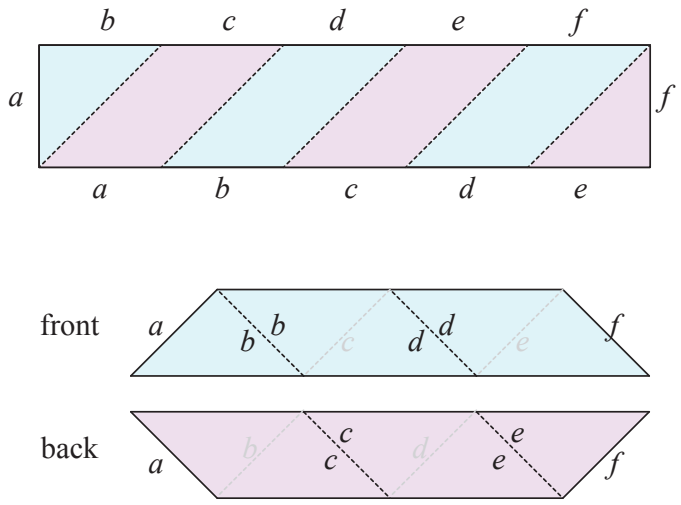

Figure 1: Top: rectangle with fold creases: n_p=4, n_g=12. Bottom: Two views of doubly covered trapezoid: n_s=7, n_c=4.

case to n_s=7. Three of these points on the trapezoid sides are not cone points, having 2π of surface surrounding them.

For the algorithm described in the next section, it is n_s that largely determines the complexity. Because n_s and n_g are linearly related, and $n_g > n_s$, it is cleanest to define $n = n_g$, the complexity of the gluing instructions. In many cases, all four n's are linearly related, but in general it could be that n_g (and so n_s) are arbitrarily larger than n_p and n_c. We must have $n_g \geq n_p$ and $n_s \geq n_c$.

The relationship between n_p and n_c is quite close. Not each of the n_p vertices of a polygon necessarily ends up as a vertex of $\mathcal{P}$, because vertices of the polygon whose angles sum to 2π can be glued together. And not each of the n_c cone points of $\mathcal{P}$ derives from a polygon vertex, because one can create a *fold point* of angle π at the interior of a polygon edge. Thus there is no exact relationship between n_p and n_c. However, there can be at most four fold-points [DO07, Lem. 25.3.1], because the Gauss-Bonnet Theorem limits the total curvature of $\mathcal{P}$ to 4π,. So we have $n_c \leq n_p + 4$.

In summary, n_g dominates all the others, so by the choice n=n_g, we have all four n's are $O(n_g) = O(n)$.

3 The Algorithm

The result of gluing the polygons together according to the gluing instructions results in $\mathcal{P}$, which could be called an *abstract polyhedral surface* [Pak10, Ex. 39.11]. $\mathcal{P}$ has zero curvature everywhere except at its cone points. Forming a data structure representing $\mathcal{P}$ can be accomplished in $O(n_g) = O(n)$ time.

The next step of the algorithm is to identify the n_c cone points of $\mathcal{P}$, which are vertices of the convex polyhedron P guaranteed by Alexandrov's Theorem. (We are using $\mathcal{P}$ for the abstract surface and P for the geometric polyhedron.) Call them $v_1, \ldots, v_{n_c}$ in arbitrary order. This step can be achieved in $O(n_c) = O(n)$ time.

The second step of the algorithm is to find the $\binom{n_c}{2}$ shortest paths on $\mathcal{P}$ from each v_i to each v_j. Call this set of shortest paths Σ. Note that here we cannot be assured we can use an algorithm for finding shortest paths on a convex polyhedron P if that algorithm uses the 3D structure of P, because we only have available the abstract surface $\mathcal{P}$. This excludes the use of the fastest known algorithm, the $O(n \log n)$ algorithm in [SS08], whose first step builds an oct-tree data structure around P in 3D. However, the Chen and Han algorithm [CH96] works entirely intrinsic to the surface, and so can be applied to $\mathcal{P}$. That algorithm assumes the surface is triangulated, and unfolds the surface triangle-by-triangle. Since triangulating a planar graph with n_s vertices results in $O(n_s)$ triangles, the appropriate count here is $n_s = O(n)$ by our choice of n in Section 2. The Chen and Han algorithm has time complexity $O(n^2)$. Repeating this for each vertex v_i as source results in $O(n^3)$ time for this step. It is possible this brute-force approach to computing all vertex-vertex shortest paths could be improved, but we make no attempt here.

A key fact we use at this juncture is that every edge of a convex polyhedron P is the shortest path on P between the two endpoint vertices it connects. This follows because any other path between those endpoints is not a straight segment in 3D, and so is strictly longer. Thus we know the unknown edges of P are among the $O(n^2)$ shortest paths Σ on $\mathcal{P}$.

If indeed P is a flat polyhedron, then it is a doubly covered convex polygon, whose *rim* ρ contains all the vertices $v_1, \ldots, v_{n_c}$. Moreover, the path on $\mathcal{P}$ that constitutes ρ must bisect the angle at each v_i, because the half-angle on one side is mirrored on the other side. So we look for such a path. We now show that:

Claim 1. If ρ exists, it can be found in $O(n^3)$ time.

Claim 2. If ρ is found, then P is uniquely identified as a flat polyhedron.

Start with v_1, and look at the shortest path $\sigma(v_1, v_j)$ to each v_j, $j > 1$ in turn. For each of these, see if it can be extended by $\sigma(v_j, v_k)$ so that (v_1, v_j, v_k) bisects the total angle at v_j. If so, this a potential start to ρ, and so this path should be followed. The path is now entirely determined by the bisection property: the path through each vertex must bisect the total angle there. If bisection holds at each step, and all vertices are included into one loop, then we have found a candidate for ρ. If at any stage, an outgoing bisecting shortest path is not available from Σ, that search can be abandoned, as it could never produce ρ. If the bisecting path closes into a loop without passing through every vertex, again we may discard it. In the case that we have found a candidate for ρ, we make one more test, for simplicity, non-self-intersection, for ρ must

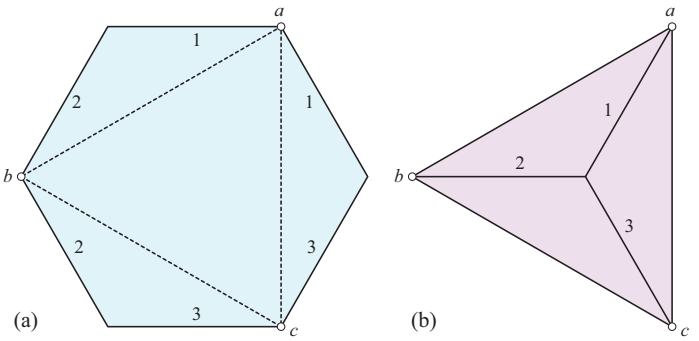

Figure 2: A regular hexagon (a) that folds (toward the viewer) to a doubly covered equilateral triangle (b). The edge labels indicate gluing instructions.

be a simple closed path. Although it is unclear if there can be a bisecting self-crossing path through every vertex, in the absence of a proof of non-existence, we simply check for this condition.

Let us illustrate before proceeding with the description. Figure 2(a) shows a regular hexagon, whose gluing instructions fold it to an equilateral triangle (b).[2] In this simple example, the bisecting path $\rho = (a, b, c)$ would be found immediately.

Figure 3 shows a slightly more complex example, based on [DO07, Fig. 25.24]. Note the identification of the four vertices of $\mathcal{P}$ in (a) of the figure. Suppose $v_1 = a$ and $v_j = c$. The path (a, c) can indeed be extended through c to bisect the angle of π there, but only by prematurely returning to a (and then it does not bisect at a). So this path would be abandoned by the algorithm. The path beginning (a, b) continues to $\rho = (a, b, c, d)$.

Returning to the argument, let us count up the worst-case complexity of following one (v_1, v_j) path, without attempting sophisticated algorithms. Let us assume the shortest paths in Σ are maintained in sorted order around each source vertex, which is easily returned by the Chen and Han algorithm. We start with $\sigma(v_1, v_j)$, and search in Σ for a bisecting extension $\sigma(v_j, v_k)$ in $O(\log n)$ time. This same cost is incurred at each successive step. We also must check at each step if we have prematurely closed a loop, which can be accomplished with a constant-time array lookup of the previously visited vertices. So a full path $\rho = (v_1, v_j, \ldots, v_1)$ can be found in $O(n \log n)$ time. Finally, we need to check ρ for simplicity. Although it seems possible this could be accomplished in $O(n \log n)$ time, or even in $O(n)$ (because determining whether a polygon is simple can be accomplished in $O(n)$ time [Cha91]), let us just count this as $O(n^2)$ by a brute-force comparison of every pair of edges.

We repeat this procedure for the $n - 1$ possible starts (v_1, v_j), and so spend $O(n^3)$ time overall either finding a ρ, or determining that no such ρ exists. If there is such a ρ, we must find it by this procedure, because it must pass through

[2]This example was used by Daniel Mehkeri in a *Math Overflow* question 4 July 10.

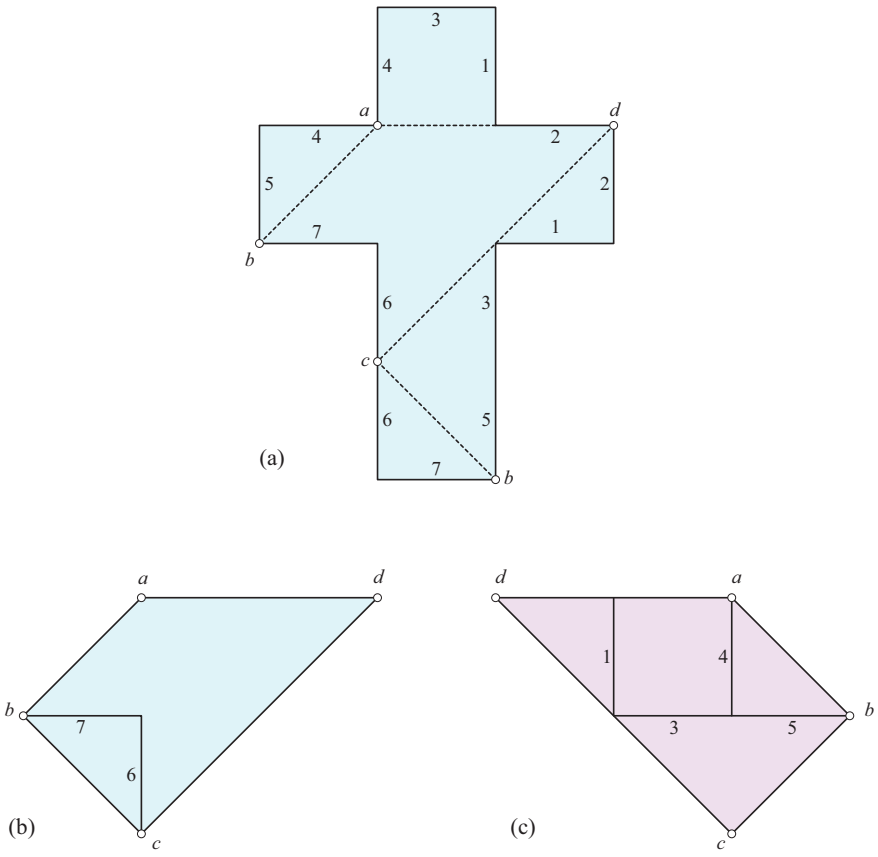

Figure 3: Folding of the Latin cross (a) (away from the viewer) to a doubly covered quadrilateral, (b) and (c). The number labels indicate gluing instruction.

v_1. So we have established Claim 1 above.

Claim 2 is that if we find ρ, then indeed P must be the doubly covered flat convex polygon whose boundary is ρ. Here we can employ this result from [IOV10, Cor. 4]:

Lemma 1 *A convex polyhedral manifold with convex boundary and with no interior curvature is isometric to a planar convex polygon.*

The proof of this lemma uses both Alexandrov's Theorem and a separate lemma of Alexandrov. Because ρ includes all vertices, the interior is indeed curvature-free, so each "half" of $\mathcal{P}$ bounded by ρ is isometric to a planar convex polygon. So we know we have a doubly covered convex polygon P. Finally, Alexandrov's Theorem establishes that P is unique, so there is no need to seek another ρ.

4 Conclusion

Although reconstructing the 3D structure of the polyhedron guaranteed to exist by Alexandrov's Theorem is a challenging problem, it is relatively easy to detect the degenerate flat-polyhedron case of the theorem, and to reconstruct the doubly covered convex polygon: Just follow vertex-to-vertex shortest paths seeking the rim. Although the algorithm described has cubic time complexity, it seems possible that it could be reduced to near-quadratic complexity.

Returning to the different n's discussed in Section 2, the time complexity is $O(n_g)$ to form the abstract surface $\mathcal{P}$, $O(n_s^2 n_c)$ to find the shortest paths, and $O(n_c^3)$ to find ρ. The factor that dominates is n_s^2, so it could be worthwhile to reduce n_s by merging coplanar faces prior to running the shortest-paths algorithm.

Perhaps a more interesting direction for future research is to explore whether other special classes of polyhedra P might be reconstructable from an Alexandrov gluing without all the machinery of [BI08].

Acknowledgments. The idea for this note resulted from a stimulating workshop conversation with Alexander Bobenko, Ivan Izmestiev, and Konrad Polthier in 2007. I thank Anna Lubiw for raising the wrapping issue discussed in Section 2.

References

[ADD+11] Common Developments of Several Different Orthogonal Boxes. Zachary Abel, Erik Demaine, Martin Demaine, Hiroaki Matsui, Günter Rote, and Ryuhei Uehara. *23rd Canadian Conference on Computational Geometry*. 2011.

[Ale05] Aleksandr D. Alexandrov. *Convex Polyhedra*. Springer-Verlag, Berlin, 2005. Monographs in Mathematics. Translation of the 1950 Russian edition by N. S. Dairbekov, S. S. Kutateladze, and A. B. Sossinsky.

[BI08] Alexander I. Bobenko and Ivan Izmestiev. Alexandrov's theorem, weighted Delaunay triangulations, and mixed volumes. *Annales de l'Institut Fourier*, 58(2):447–505, 2008.

[CH96] Jindong Chen and Yijie Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.

[Cha91] Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[DO07] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, July 2007. `http://www.gfalop.org`.

[IOV10] Jin-ichi Itoh, Joseph O'Rourke, and Costin Vîlcu. Star unfolding convex polyhedra via quasigeodesic loops. *Discrete Comput. Geom.*, 44:35–54, 2010.

[O'R07] Joseph O'Rourke. Computational geometry column 49. *Internat. J. Comput. Geom. Appl.*, 38(2):51–55, 2007. Also in *SIGACT News*, **38**(2): 51–55(2007), Issue 143.

[Pak10] Igor Pak. Lectures on discrete and polyhedral geometry. `http://www.math.ucla.edu/~pak/book.htm`, 2010.

[SS08] Yevgeny Schreiber and Micha Sharir. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Discrete & Comput. Geom.*, 39:500–579, 2008.