3.2.1 Cut-tree truncation and the smallest undevelopable polyhedron

It is interesting to conjugate the previous conjecture with the empty sector property. Consider the example given above: though the truncated vertex was a leaf-node of the graph, it did not have the empty sector property. If it had, then by Theorem 2.5 there would have been no overlap.

This shows that

Lemma 3.2 *If there exists a smallest ununfoldable polyhedron P_0 and P_0 has triangular faces, then every possible polyhedron and cut-graph (P_1, T) which can be leaf-truncated to produce P_0 must lack the empty sector property at the vertex of truncation.*

Proof. Let P_0 be the smallest ununfoldable polyhedron.

Assume that P_0 has triangular faces. Label one such face $\triangle abc$.

Let P_1^{abc} be the particular polyhedron whose vertex v must be truncated to generate P_0 with face $\triangle abc$.[2]

By assumption on P_0, P_1^{abc} is developable. The only difference between P_1^{abc} and P_0 is the removal of v and its replacement with abc, so the location in the plane of every unfolded face in P_1^{abc} will be the same in the unfolding of P_0. This means that any conflict (and its is known that there is a conflict) must come from $\triangle abc$.

Furthermore, this is a conflict which will appear *for every possible valid cut-graph of P_1^{abc}*, because if there existed a single cut-graph which could unfold P_1^{abc} in such a way that the truncation of $v \rightarrow \triangle abc$ produced no overlap then P_0 would be unfoldable.

Thus if P_0 is the smallest ununfoldable polyhedron and P_0 has triangular face $\triangle abc$ achieved by truncating the vertex v from the smaller polyhedron P_1^{abc}, then the vertex v does not have the empty sector property under *every possible cut-graph which unfolds P_1^{abc}*. $\square$

It is somewhat difficult to categorize this result. It does not eliminate the possibility that there is an undevelopable polyhedron with triangular faces, and in this sense it is quite weak; and yet, it places an almost absurdly stringent requirement on the conjectured surface, one which would seem intuitively to be almost impossible to meet, and in this sense the lemma is surprisingly strong.

3.3 Angular Restrictions to Ensure Developability

In (Luc06, p.54), Brendan Lucier illustrates the angular requirements for *2-local overlap*, the simplest possible form of self-intersection in the unfolding of a convex polyhedron. Lucier shows that if a branch of the cut-graph 'curls' too much upon itself then overlap becomes possible. Lucier's goal was to demonstrate how to create overlap, but it becomes interesting to turn the argument around

[2]Note the assumption that there exists exactly one such P_1^{xxx} for each triangular face $\triangle xxx$ in P_0. This is not an accurate assumption, for the reasons detailed above in the first objection to the conjecture (that is to say, P_1^{xxx} may not exist at all) but this inaccuracy is immaterial to the substance of the discussion, and will be disregarded.

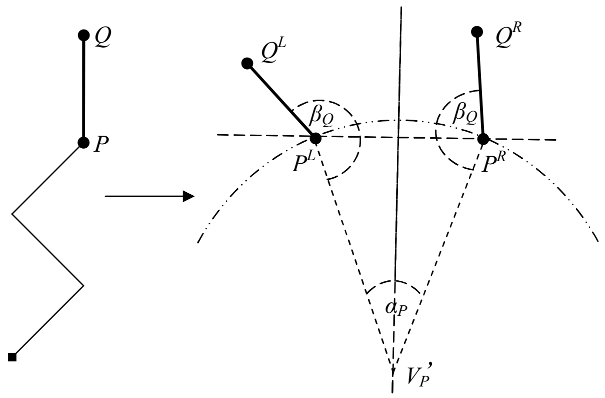


Figure 3.3: Unfolding the branch Φ. The angle between P^LQ^L and P^RQ^R is α_Q. The lesser of the angles from V_P' to each of the developments of PQ is β_Q.

and ask: is there an upper bound on how sharply a cut-graph branch can turn, below which there is no risk of self-intersection?

This section presents an extension of the so-called 'Alpha-Beta Rules', a set of lemmas originally presented to the Sixth International Conference on Curves and Surfaces, Avignon 2006 as *Angular Restrictions to Ensure Developability* (Ben06). The original work was, unfortunately, done before the author became aware of Lucier's research, and in their original form the alpha-beta rules were strictly limited by the particular geometry of the surface to unfold. The material presented in this section represents a step beyond those restrictions.

At Avignon, the alpha-beta rules were described as being, "designed to show that it is possible to 'grow' a cut-graph from the outermost leaves inwards to form a complete, spanning tree guaranteed by its incremental assembly to unfold without overlap.". Strictly speaking, this was a misleading summary: there is no 'time' in cut-graphs, thus no growth. A more accurate description of the intent of the alpha-beta rules would have been to say that they are designed to provide a *partial ordering of the vertices of the polyhedron*, established by a set of sequential choices, such that any unfolding which follows this ordering will be free of overlap.

This section presents the building blocks of a proof. The proof is not complete, and the work which remains to be done is detailed before the end of the section.

3.3.1 Terminology

Given the nature of the task, much of the following discussion will revolve around some partially-formed cut-graph and the task of selecting which vertex to add to it next. It will be assumed that the leaves of the cut-graph are 'fixed' and that each new node to discover will be a step away from the leaves of the graph and toward the root; the reader might picture a set of streams, emerging from sources around the slopes of a valley and gradually merging together into larger rivers. Thus for any subset of the cut-graph, there will be a single vertex which marks the current 'river mouth', with the path downstream as yet unknown.

As the phrase 'subset of the cut-graph' is cumbersome to wield, the shorter term *cut-path* will be used here to describe any connected subset of some cut-graph T which is to be discovered on a convex polyhedron. In the original publication of (Ben06) a cut-path was limited to being a connected and Hamiltonian path, but that restriction is lifted here. A cut-path may contain multiple leaf nodes of T. All leaf nodes of a cut-path are treated as fixed, and each cut-path has a single vertex which is the *tip* of the cut-graph. The *tip* is the node the furthest from the leaf nodes; all growth of a cut-path, and union with other cut-paths, occurs at the tip. In this section, the cut-path currently being discussed will be labeled 'Φ' and its tip will be labeled 'P' (Figure 3.3.)

Let α_P be the sum of the angle deficits of the vertices of Φ, up to *but not including* the angle deficit of P. Recall that the angle between the two developments of any edge is exactly the sum of the angle deficits of all cuts which lead to that edge, and that the total angle deficit of any surface of genus zero is 4π. α_P is not defined if P is a leaf node of Φ (that is, if Φ consists of a single vertex.)

Let the *transverse line* of P be defined as the line which passes through the points P^L and P^R.

The goal will be to find a new vertex Q to add to Φ beyond P, 'downstream' of P. The intent is that the developments of edge PQ should not cross each other, nor cross the developments of any other edge upstream in Φ.

Let β_Q be the turn angle between the cut-path and the developments of PQ. To speak of the 'turn angle' from what could easily be a tree with many leaves is quite imprecise, and so a new concept is introduced to refine the definition of β: the *virtual root*.

Note: In each of the following lemmas, it is assumed that α_P is less than π. The case where $\pi \leq \alpha \leq 4\pi$ is discussed below.

3.3.2 The virtual root

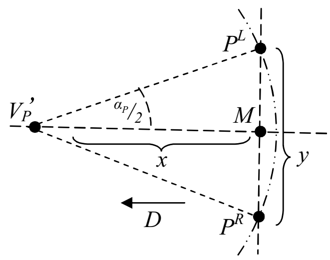
When unfolded, P will develop to two distinct points in the plane, P^L and P^R. These points define a *virtual root*, V'_P, an idea first encountered as 'collapsing a fork' in the discussion of locally-convex cut graphs in Section 2.6 . V'_P is defined to be the point[3] in the plane which lies on the perpendicular bisector of $P^L P^R$ to the right of the ray $P^L \to P^R$ at a distance d from P^L and P^R chosen such that the angle between the line segments $V'_P P^L$ and $V'_P P^R$ is exactly α_P.

Given V'_P, for any vertex Q which is being considered for addition to Φ beyond P, β_Q is defined to be the lesser of the two angles $\beta^L_Q = \angle V'_P P^L Q^L$ (measured counterclockwise) and $\beta^R_Q = \angle V'_P P^R Q^R$ (measured clockwise.) Note that β_Q is always positive.

In many developments of an edge, one edge will 'turn away' further than the other. Formally, if β^L_Q is less than β^R_Q then the *inner development* of PQ

[3] Calculating V'_P:

$$
\begin{aligned}
M &= (P^L + P^R)/2 \\
y &= \|P^L - P^R\|/2 \\
D &= ((P^L - P^R) \times [0,1,0])/y \\
x &= (y/2)\cot(\alpha_P/2) \\
\\
V'_P &= M + xD
\end{aligned}
$$



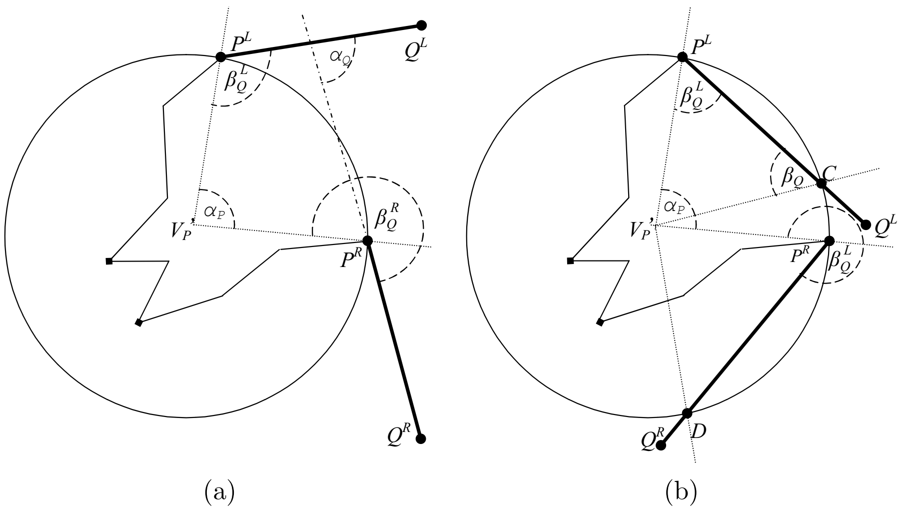


Figure 3.4: If $\beta_Q > \frac{\pi-\alpha_P}{2}$ then there can be no conflict between the left and right developments of PQ. (a) $\beta_Q \geq \frac{\pi}{2}$ (b) $\frac{\pi-\alpha_P}{2} < \beta_Q < \frac{\pi}{2}$

is P^LQ^L and the *outer development* of PQ is P^RQ^R; and vice-versa, if the comparison of the angles is reversed. The inner development of PQ may be denoted by P^IQ^I.

Let Γ be the circle of radius d centered on V_P', passing through P^L and P^R.

3.3.3 Preventing overlap within each pair of unfolded edges

Lemma 3.3 (Alpha-Beta Rule 1) *Given a cut-path Φ with tip vertex P, virtual root V_P', and a potential extension vertex Q to be considered for addition to the cut-path beyond P, if $\beta_Q > \frac{\pi-\alpha_P}{2}$ then it is impossible for P^LQ^L and P^RQ^R to intersect.*

Proof.

Lemmas One and Four of (Ben06, p.3) gave a detailed mathematical proof that if $\beta_Q \geq \frac{\pi-\alpha_P}{2}$ then P^LQ^L and P^RQ^R cannot cross; the arguments presented in the paper hold just as well for V_P' as for V. The proof given was difficult to interpret and so a more geometric argument is offered here.

Case (a) — $\beta_Q \geq \frac{\pi}{2}$:

Consider Figure 3.4(a). If $\alpha_P = 0$ then the two developments would be coincident, but on a convex surface $\alpha_P > 0$. As α_P increases, the distance from P^L to P^R must increase by the length of the chord of arclength α_P on Γ; simultaneously, Cauchy's Arm Lemma (p. 18) dictates that as the angle α_P grows, the distance from Q^R to Q^L must increase. Thus (up to $\alpha_P < \pi$) the two developments can never cross.

Case (b) — $\frac{\pi-\alpha_P}{2} < \beta_Q < \frac{\pi}{2}$:

If $\beta_Q < \frac{\pi}{2}$ then the edge P^LQ^L intersects Γ at some point C (Figure 3.4(b).) The triangle $\triangle P^LCV_P'$ is an isosceles triangle with angles β_Q, β_Q, $\pi - 2\beta_Q$.

The angle of the triangle at V_P', between the lines $V_P'P^L$ and $V_P'P^R$, is α_P. The angle deficit of P itself does not factor into the angle between the developments leading up to P.

If $\alpha_P > \pi - 2\beta_Q$ then P^R must lie outside the triangle $\triangle P^L C V'_P$, in which case the triangle $\triangle P^R D V'_P$ does not overlap $\triangle P^L C V'_P$ at any point other than V'_P.

Therefore if $\beta_Q > \frac{\pi - \alpha_P}{2}$ then $P^L Q^L$ and $P^R Q^R$ cannot intersect. $\Box$

Corollary 3.4 *Given a cut-path Φ with tip P and potential extension vertex Q, the angular restrictions to avoid overlap between the developments of PQ apply only to the inner development, which has the lesser turn angle in the plane. The remaining development may turn freely: $AD(P)$ is irrelevant to the potential overlap of the developments of the edge PQ.*

Proof. β_Q is defined to be the minimum of the turn angles of the left and right developments, measured counterclockwise and clockwise respectively; label these two angles as β^L_Q and β^R_Q. Lemma 3.3 guarantees that if $min(\beta^L_Q, \beta^R_Q) > (\pi - \alpha)/2$ then there is no risk of overlap between the developments of this edge, leaving $max(\beta^L_Q, \beta^R_Q)$–which determines the position of the outer development–unconstrained. $\Box$

Corollary 3.5 *Given a cut-path Φ with tip P and potential extension vertex Q, if $\beta > \frac{\pi - \alpha_P}{2}$ then the developments of PQ will not cross the 'upstream' developments incident to P^L and P^R.*

Proof. Recall that it has been assumed that $\alpha_P < \pi$, placing V'_P to the right of the ray $P^L \to P^R$. By the same token, the two edge-developments leading to P^L and P^R must also lie to the right of $P^L \to P^R$. The proper choice of Q must either place the developments of PQ to the left of $P^L \to P^R$, or to the right of $P^L \to P^R$ but outside the line segment $P^L P^R$. Thus there can be no intersection between the developments of PQ and the other two developments incident to P^L and P^R. $\Box$

3.3.4 Extending a cut-path

Having shown that there are broad angular constraints within which an edge can be cut without risk of overlap, it can now be shown that an edge which meets these constraints must always exist:

Lemma 3.6 *Given a cut-path Φ ending in tip vertex P, there must exist at least one vertex Q such that $\beta_Q > \frac{\pi - \alpha_P}{2}$ for the edge PQ.*

Proof.

Consider Figure 3.3, in which the transverse line through $P^L P^R$ divides the plane in two; call the side containing V'_P the 'upstream' side, opposite the 'downstream' side. Recall that the angle between any two developed edges which share a common developed vertex is the sum of the face angles between them.

Then if there were no edges in the downstream half of the plane (Figure 3.5) it would imply that two edges met at P^L or P^R with no edge of the polyhedron between them but also with more than π radians of incident face angle. This would mean that the face shared by the two edges had $> \pi$ incident angle at a vertex, which is impossible on a convex polyhedron. No face of a convex polyhedron may itself fail to be convex.

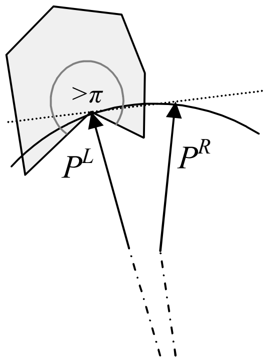


Figure 3.5: There must exist at least one vertex with $\pi/2 \leq \beta < 3\pi/2$; for this not to be the case would imply that a face incident to P was nonconvex.

If there is an edge in the downstream half of the plane, β for that edge would be between $\pi/2$ and $3\pi/2$. Even if α were zero, this would still be sufficient to guarantee that there is no overlap in cutting that edge. □

3.3.5 Preventing overlap between an edge and previous edges on the cut-path

Lemma Two of (Ben06, p.4) proved that, under certain stringent conditions, each new edge added to a cut-path would not cross *any* development of an edge already in the cut-path. Unfortunately, those conditions were too strict: the lemma required that (a) Φ was Hamiltonian and thus had exactly one root, labelled V; and (b) that that root was coincident with the virtual root V'_P.

Thus far, Lemmas 3.3 and 3.6 and the associated corollaries have shown that if Q is chosen to have $\beta_Q > \frac{\pi - \alpha_P}{2}$, then PQ will cross neither itself nor its immediate predecessor in the cut-path, and that there must always exist at least one Q which fits the bill.

However, the restriction on β is not sufficient to guarantee that there will be no overlap with prior edges in the cut-path. This is because the restriction on β is strictly local; to avoid overlap there must also be a global factor guiding the choice of edge. Inspired by the design of the Anchorable Convex Hull, what is needed is some central anchoring point or face.

The *center of the unfolding*, C, is an arbitrarily-chosen point, somewhere within the border of the net, which will determine a focal direction for the unfolding. The center of the unfolding may be chosen from the development of any point on the polyhedral surface. In the software implementation of the alpha-beta rules in YAMM, the author has chosen to use the center of the development of the lowest face in the polyhedron, but any point which falls within the first developed polygon will do.

Lemma 3.7 (Alpha-Beta Rule 2) *Let Φ be a cut-path with tip P, where every edge already in Φ has been chosen according to the constraint presented in this lemma. Let Q be a potential extension vertex to Φ, and label the inner development of PQ (be it the left or right development) as $P^I Q^I$. Let C be the*

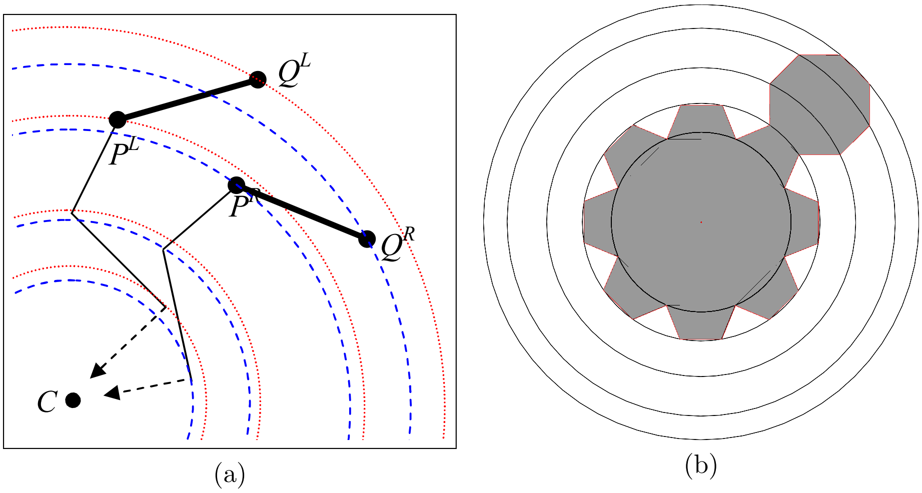


Figure 3.6: (a) Each development from $P \to Q$ is bounded by concentric rings around V (left-hand developments are shown on red dotted lines, right-hand on blue dashed lines) (b) Concentric rings around the center of the radially-oriented unfolding of the shallow truncated octahedral pyramid

center of the unfolding, a point within the development of the root face of the unfolding. Then, if Q is chosen such that

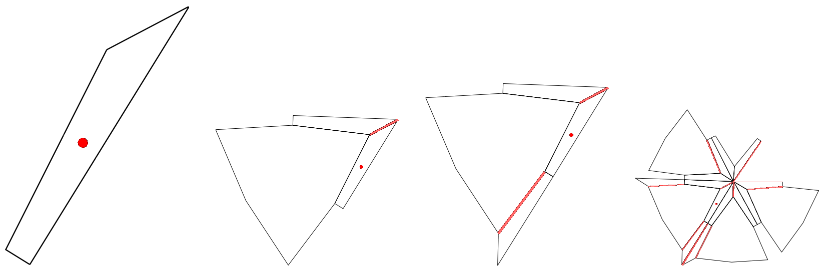
- $\beta_Q > \frac{\pi - \alpha_P}{2}$
- *Q maximizes the value of the dot product $\frac{CP^I}{\|CP^I\|} \cdot \frac{P^I Q^I}{\|P^I Q^I\|}$ above any other vertex in the 1-ring of P*
- *The dot product is greater than zero*

then the left-hand development of PQ will not cross any left-development of any edge already in Φ, and the right-hand developments likewise will not cross.

Proof. Each new vertex added to Φ will be chosen such that the vector from the inner development of P to the associated development of Q is a vector which points, as nearly as possible, directly away from the center. It has already been established that at least one Q must exist; this is a criterion for distinguishing between multiple valid options, should they be available.

The inner development of each new vertex added to Φ will always fall at a greater distance from C than its predecessor if $\frac{CP^I}{\|CP^I\|} \cdot \frac{P^I Q^I}{\|P^I Q^I\|} > 0$ (Figure 3.6(a).) If each new vertex develops to progressively further radii from C then they can never cross a preceding edge. $\square$

By the same logic as that used in Lemma 3.6, there must always be at least one Q which would travel away from C, although it is less clear that such a Q would also comply with the rules on β.



(a) Step zero (b) Step one (c) Step two (d) Step eight

Figure 3.7: Four steps in the discovery of a cut-path on a banded icosahedron. The red dot marks the root face.

3.3.6 Cut-paths and partial unfoldings

Before proceeding, further discussion of what it means to 'cut an edge' is warranted. In the scenarios discussed here, the developed edges of a cut are often treated as simple joined line segments in the plane; but the reader must recall that they are in reality the outer edge of a polygon which has been unfolded in its entirety. To cut an edge is to partially determine the relative positions of the faces nearby, which can be exploited when developing a surface. This insight has interesting implications.

For example, consider a leaf-node of the cut-graph. Because only one edge incident to the leaf-node is cut, the fan of faces around the vertex are locked together; their relative placement is fixed and immutable.

Likewise, consider a vertex cut by two incident edges. The two arcs of faces separated by the cuts are still bound together within themselves; the positions of all the faces to the left of the cut are fixed relative to each other, and likewise for the right.

Recall from p. 11 that a *partial unfolding* U is a connected subset of the unfolded faces of a polyhedron. Each cut-path Φ defines a unique partial unfolding U_Φ: U_Φ is the unfolded net of faces which are incident to a vertex in Φ. The edges of Φ are a subset of the perimeter edges of U_Φ.

To find the partial unfolding U_Φ determined by a cut-path Φ, the partial unfolding is constructed iteratively. Begin with any face f_0 incident to any vertex V_0 which is a leaf-node of Φ. Develop f_0 into the plane; this will be the root face of U_Φ. Thereafter, for every face f which is developed, for each undeveloped face g which is adjacent to f across a shared edge $e \notin \Phi$ where g is incident to a vertex of f which is also a node of Φ, develop g by gluing its image to the development of e.

This process is illustrated in Figure 3.7, which shows several early steps in the discovery of cut-paths to unfold a banded icosahedron. Faces are added as each new vertex is expanded; in time, the separate cut-paths will merge together to form a single complete cut-graph.

3.3.7 Vertex ordering and interdependency

The lemmas presented thus far have focused on the extension of a single cut-path from a single tip. However, such focus may be misleading, because vertices are not extended in isolation. Consider: to choose the best outbound edge to cut from a given vertex P, α and β must be calculated. To find α requires prior knowledge of every edge which will be cut up to P in the cut-path. Thus the expansion of P is dependent on the prior expansion of every other vertex in the cut-path before P.

For example, what would it mean to cut an edge from P to Q if Q has already been expanded? Doing so changes the value of α_Q at Q by adding new incident cut edges. This in turn could mean that some new edge which had previously not met the constraints on β_Q is now a candidate for expansion, perhaps pointing more directly away from the center. Then Q's expansion must be recomputed, which could mean re-expanding every vertex past Q in the cut-path.

The solution is two-fold: *sort vertices* by increasing distance from the chosen center point, and support a limited amount of *backtracking*.

Sorting vertices for expansion

The goal in sorting vertices by distance is to minimize–or, if possible, eliminate–the number of occasions where a vertex is expanded by cutting to a vertex which has already been expanded. As every expansion will point away from the center according to Lemma 3.3, it makes sense to first choose vertices nearest to the center. Assume that the unfolding has begun at the lowest face of the polyhedron F_0 and that the center point C of the unfolding has been chosen to be the center of the development of F_0. There are then several sorting options available:

1. Sort vertices lexicographically with height as the primary axis.
2. Sort vertices by geodesic distance from the center of F_0, breaking ties with lexicographic order.
3. Sort vertices by chord-length inside the polyhedron (linear distance in space)
4. Sort vertices by the minimum of the radii of the developments of each vertex
5. Sort vertices by the maximum of the radii of the developments of each vertex

After fairly extensive experimental trial and error, the author has chosen option (5) for the YAMM software, scoring each vertex by the maximum of the radii of all of its developments and then sorting all vertices by increasing score, developing the lowest-scored vertex first. It was found that this method yielded the lowest error rate.

To implement this sorting method, the vertices of the polyhedron cannot be scored ahead of time: the radii to which a given vertex develops depend completely on the developments of prior edges on the cut-path. However, this is acceptable: with each step of the algorithm, only the nearest unexpanded vertex needs to be identified, and there will always be unexpanded vertices on the outside border of the partially-unfolded net until the net is complete.

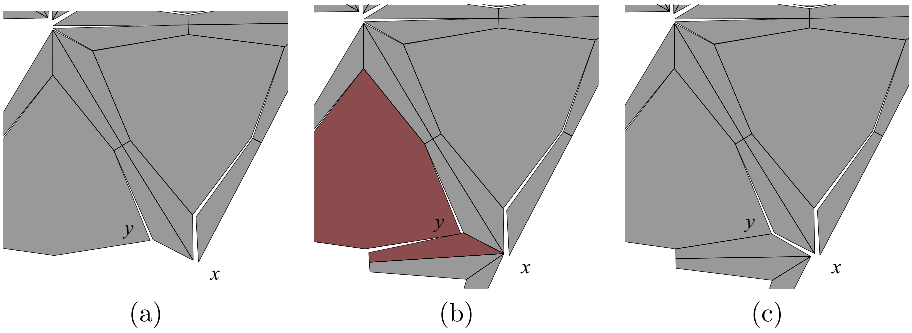


Figure 3.8: Three sequential frames from the unfolding of a banded icosahedron by the alpha-beta rules. In the course of expanding vertex x from frame (a) to frame (b), several faces are unfolded to the plane which break the β rule at vertex y. y has not yet been developed; if it had, the edge xy would already have been cut, and the expansion of x could not have restored it. From frame (b) to frame (c), y is expanded and the error is automatically rectified: the edge xy is now part of the cut-graph.

Limited backtracking

Experiment has verified that although sorting method (5) minimizes the average incidence of cuts ending at already-expanded vertices more effectively than any of the other options, it does not eliminate it. This is demonstrated in Figure 3.8, where a partial unfolding grows into–and then out of–self-intersection while following the alpha-beta rules.

In Figure 3.8(a) the vertex x is about to be expanded. The best outbound edge is found, cutting downwards and to the left in the figure (the center, not shown, is upwards and to the right.) In Figure 3.8(b) the expansion of x has added three more faces to the net, creating a less-than-$\pi/2$ turn in the cut-path around adjacent vertex y which has not yet been expanded. In Figure 3.8(c) the ordering of expansion has reached y and the best (in fact, only) edge which can be cut from y is the edge xy; β at y returns to a valid value. The sole concern is that in expanding y after x, y's new cut has altered the total α feeding into x. x should now be re-expanded, although in this case (and in the majority of cases, during testing) this re-expansion will not change the choice of outbound edge.

This demonstrates that there is an *interdependence* between some sets of vertices: situations where one vertex must be expanded before another, even though the distance-based ordering would address them in the wrong order.

This dependence relation could presumably be expressed as a dependency graph between vertices. However, this graph would change from step to step as cut-paths were added or merged; it seems improbable that a polyhedron-wide 'dependency map' could be built for a surface without constructing the unfolding beforehand. One version of the YAMM software implementation of the alpha-beta rules explored this possibility. At every step the algorithm would simulate every possible expansion of each of the vertices on the rim of the partial net and then choose to expand only from amongst vertices whose 'best' expansions would not develop faces across edges which another vertex would later need for its own 'best' expansion. In testing, this subroutine proved to be cumbersome, difficult to extend, and insufficient to the task; it only provided 'lookahead' for one edge, where even the banded icosahedron has vertices which are interdependent across two or even three adjacent edges.

Rather than take the software down the fruitless path of unlimited-depth predictive search, a simpler solution was instead chosen, which has yielded the frames shown in Figure 3.8. For every vertex, always cut the 'best' edge according to the rules for β and C, even if that edge was previously laid down unbroken by another vertex's expansion; but if such an edge must be cut, then backtrack one step and re-expand the impacted second vertex.

Although it will not be formally proved, it seems reasonable to argue that this backtrack operation must terminate locally without inducing an uncontrolled backtrack cascade. This is because each vertex which is re-opened for backtracking will still cut an edge to point outwards, away from C. The newly-cut edge should not touch the vertex which caused the backtrack.

3.3.8 When α is greater than π

Throughout this discussion of the new alpha-beta rules, three quarters of the problem have actually been ignored: the rules above are for when α is less than π, but the total angle deficit on a closed convex polyhedron is 4π. What of the rules for $\pi \leq \alpha \leq 4\pi$?

It is tempting, in all honesty, to dismiss the question. Consider: unfolding will begin at the lowest face, with a cut-path extending from each vertex, so there are already at least three cut-paths from the very first step. If they stay separated until the top of the surface these cut-paths will each account for (on average) $4\pi/3$ radians' worth of total angle deficit; one more cut-path independently reaching the top and that average drops to π and the rules above apply.

Another reason to disregard $\alpha > \pi$ is that overlaps occur so rarely for higher values of α. The vast, vast majority of overlaps are local events, n-local collisions for small n; these collisions are overlaps within cut-paths which are small parts of much larger unfoldings. By the time α has grown to π the two sides of the development are across the unfolding from each other; clashes between them could not happen without crossing all of the intervening faces first.

And yet, neither of these objections is sufficient support for arguing that higher values of α can be disregarded entirely–not if the goal is a robust proof. To complete the proof, the rules given above must be extended to greater α, but this will not be attempted in this dissertation.

3.3.9 Weaknesses in the argument

The arguments which have been presented here are not sufficient to constitute a formal proof that the alpha-beta rules can unfold every convex polyhedron. The author deeply regrets that the proof remains incomplete.

The known flaws in the argument are:

- The reasoning that there must exist an outbound edge which satisfies both the rules for β and for C is tenuous: it holds up well for Hamiltonian cut-paths but it is, in the author's opinion, not yet strong enough for the case where two or more edges of a cut-path flow into a single vertex to be expanded. Consider the volcano unfolding of a cone: the transverse line through the developments of the tip are poorly defined, but worse, the transverse line between any two non-adjacent developments is crossed by the development of the intervening face.

- Malcolm Sabin has provided an argument[4] that without Lemma 3.7, Corollary 3.5 must also require that the outer development of PQ fall to the right of the transverse line $P^L \to P^R$.

- Lemma 3.7 is actually very limited: it guarantees that there will be no overlap between developments *on the same side* but not between left and right sides. Without this, the proof is incomplete. The argument seems feasible: there is a relationship between the ordering of the left-hand and right-hand (red and blue in Figure 3.6) concentric circles which bound the developments of each new edge. It may be possible to exploit this relationship to use Lemma 3.3 and Corollary 3.5, perhaps by showing that there will never be more than two sequential red rings before the next blue (or vice versa).

- The reasoning that a vertex Q can be added to extend a cut-graph without overlap does not show that a second edge can be cut from some other cut-path to Q without causing intersection between the two cut-paths.

- As described above, the ordering of vertices is still a topic of open research. It would be nice to be able to justify the choice of the particular ordering chosen with a better reason than ‘experimental testing’.

- A significant improvement to the backtracking support would be to undo any face developments as part of a backtrack, in addition to flagging downstream vertices for re-appraisal. The difference would be that those downstream vertices had already developed their faces and those faces then influenced the developments of other nearby vertices; all of those vertices should be rolled back too.

- Formal support for $\alpha \geq \pi$ is essential.

- Despite the author’s assurance that backtrack could not induce an infinite loop, this is not actually quite true: on very rare occasions in testing, the author found that the *topmost* face of the surface would circle around the outer rim of the unfolding, looping forever as one of its vertices was developed and the other two marked for re-development continually. This is a software glitch, of course, but it does highlight the need for better-defined termination guides in the rules.

- There are still a few very rare and hard-to-categorize cases where the software will fail to find an overlap-free unfolding, despite following the rules above. These cases seem to always be related to the poorly-defined termination conditions and issues related to backtracking.

[4]Personal communications, 2008

3.3.10 Experimental support

The alpha-beta rules have been implemented in the YAMM testbed with remarkable, but not perfect, success. The algorithm can unfold extremely 'difficult' surfaces such as the banded third-level subdivision of an icosahedron; which is notable because it does so *entirely without testing for collisions*. This really is a striking result, all the more so because overlap *is* generated in the course of the unfolding–but it is repaired as the process continues. In short, the nets are clearly being generated according to rules which do prevent overlap.

However, there are still difficulties in the software with the final steps of the unfolding. As the end of work draws near on this dissertation, the final YAMM implementation of the alpha-beta rules has a success rate of 202,376 successful unfoldings out of 203,556 randomly-generated simplicial convex polyhedra, roughly uniformly distributed from 4 to 300 faces. That gives a success rate of 99.91%: impressive, but not ideal. Analysis of the failure cases suggests that the flaw lies in the interdependency of vertices in the penultimate stage of unfolding.

Algorithm A.20, a formal algorithmic declaration of the implementation of the alpha-beta rules, can be found in Appendix A on page 175.

Figure 3.9 shows three successful unfoldings.

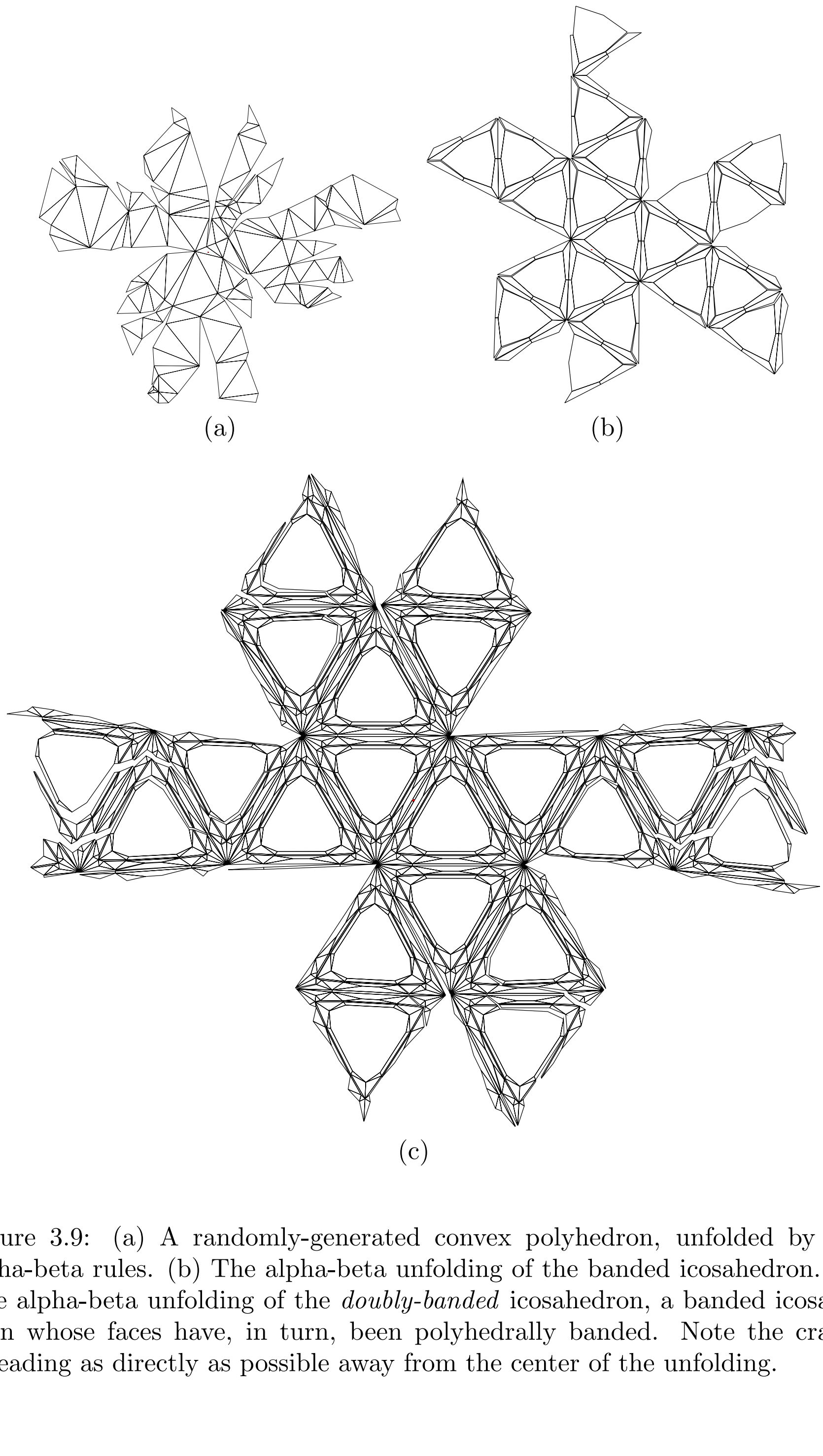


Figure 3.9: (a) A randomly-generated convex polyhedron, unfolded by the alpha-beta rules. (b) The alpha-beta unfolding of the banded icosahedron. (c) The alpha-beta unfolding of the *doubly-banded* icosahedron, a banded icosahedron whose faces have, in turn, been polyhedrally banded. Note the cracks spreading as directly as possible away from the center of the unfolding.