Proceedings of IMECE2004
2004 ASME International Mechanical Engineering Congress and RD&D Expo
Anaheim, California, USA, November 13-19, 2004

IMECE2004-62227

DETERMINING MOLDABILITY AND PARTING DIRECTIONS FOR POLYGONS WITH CURVED EDGES

Sara McMains*
Department of Mechanical Engineering
University of California, Berkeley
Email: mcmains@me.berkeley.edu

Xiaorui Chen
Department of Mechanical Engineering
University of California, Berkeley
Email: xrchen@me.berkeley.edu

ABSTRACT

We consider the problem of whether a given geometry can be molded in a two part, rigid, reusable mold with opposite removal directions. We describe an efficient algorithm for solving the opposite direction moldability problem for a 2D "polygon" bounded by edges that may be either straight or curved. We introduce a structure, the *normal graph* of the polygon, that represents the range of normals of the polygon's edges, along with their connectivity. We prove that the normal graph captures the directions of all lines corresponding to feasible parting directions. Rather than building the full normal graph, which could take time $O(n \log n)$ for a polygon bounded by n possibly curved edges, we build a summary structure in $O(n)$ time and space, from which we can determine all feasible parting directions in time $O(n)$.

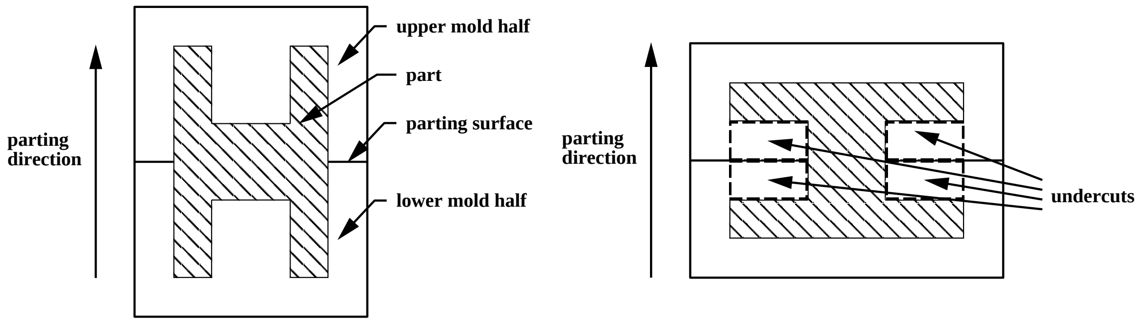
1 Background and Previous Work

In molding or casting manufacturing processes, material is reshaped in a hollow mold. A simple reusable mold consists of two rigid halves that are removed in opposite directions; the orientation of the removal directions is called the parting direction. In order for a part geometry to be de-moldable, it must be oriented relative to the parting direction so that the two mold halves can be removed from the part via translation along the parting direction without colliding with the part. Surfaces where collisions occur, preventing extraction of the part, are called undercuts. They occur where the mold extends into the area between the part and the parting surface, relative to the parting direction (Figure 1). Forming undercuts requires additional mold inserts which increase the cost of the mold, so we would like to avoid them if possible. Finding a feasible two part molding orientation (one without undercuts) for an arbitrary geometry is subject to geometric accessibility constraints; not all geometries admit such an orientation.

Many manufacturing researchers who have implemented algorithms for finding a feasible parting direction for a two part mold for a given 3D geometry (possibly defined by curved faces) only look at a limited number of potential parting directions, such as the three principle axes [1, 2], bounding box axes [3], or use a heuristic search approach [4, 5]. If a valid parting direction exists but is not among the directions tested, these algorithms will not find it. Multi-piece and/or sacrificial mold design, where constraints on the number of mold pieces and/or demoldability are relaxed, has been studied by [6], [7] and [8], but these molds are more expensive than two part molds, and sacrificial molds are not as suited to mass production.

In the computational geometry literature, algorithms for finding if *any* feasible two part mold orientation exists have been presented, but only for faceted geometry. Rappaport and Rosenbloom describe an $O(n)$ time algorithm for determining if a 2D polygon can be made by a 2-part mold with arbitrary (not necessarily opposite) removal directions for the mold halves, and an $O(n \log n)$ time algorithm for a variation on the opposite removal direction problem, again for a 2D polygon [9]. Ahn et al. describe an algorithm for opposite direction mold removal parting line direction determination for a 3D faceted polyhedron, returning all combinatorially distinct feasible directions in time $O(n^4)$ [10]. Although this algorithm could of course be applied to find feasible mold removal directions for a tessellated approximation of a curved geometry, the running time would increase prohibitively the more accurately the approximation fit the original geometric surface.

*Corresponding author.



(a) Mold for a simple 2D part

(b) An orientation of the same part with undercuts

Figure 1. *Terminology*

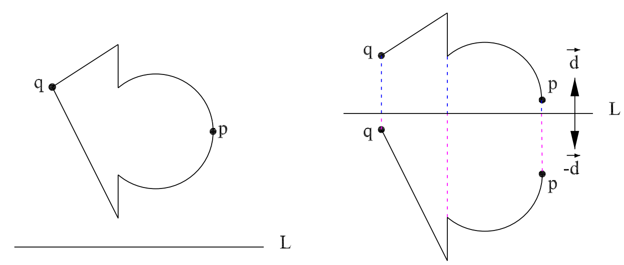
In this paper, we describe a new algorithm for solving the opposite direction moldability problem for a 2D "polygon" bounded by curved edges. Since CAD programs build 3D solid models based on Boolean combinations of extrusions, sweeps, and lofts of 2D contours, an efficient algorithm for determining the moldability of those 2D input contours can serve as a foundation for 3D moldability algorithms. Our algorithm finds all feasible parting directions for a two part mold with opposite removal directions for the given geometry. Although some geometries can be made by two part molds only when the removal directions for the two halves are non-opposite, in industrial practice, opposite removal directions are required. Thus our algorithm addresses a more industrially relevant variant of the 2-part moldability problem than [9], which looked at arbitrary mold removal directions and was not extensible to curved polygons. The running time of our algorithm is $O(n)$, where n is the number of "segments" bounding the input polygon, where we consider either a straight line segment or a curve with G^1 continuity and positive or negative signed curvature everywhere to be a segment. For input defined by spline curves with a total of n control points, the running time is thus linear in the number of control points.

2 Definitions and Problem Approach

Our input is a 2D polygon P bounded by straight and/or curved edges. We assume that P is simple (i.e. it has no holes) since only simple polygons are moldable in two part molds without cores; alternately, if we wished to determine the parting direction for a two part mold for the outer boundary only of a non-simple polygon we would take only its outer boundary contour as input. The polygon boundary ∂P is defined by a finite, ordered sequence of non-self-intersecting oriented edges, each of which intersects only the two adjacent edges in the sequence and only at their shared endpoints. We will use the right-hand rule convention that the edges of the polygon are oriented such that the interior of the polygon lies to their left, i.e., the edges are oriented and sequenced in counterclockwise order around the polygon.

A polygon P is *2-moldable with opposite mold removal directions* if ∂P can be partitioned into exactly two pieces that can be translated to infinity in some directions $\vec{d}$ and $-\vec{d}$ respectively without collision with the interior of P. The direction $\vec{d}$, or equivalently $-\vec{d}$, is called the *parting direction*.

In [9], polygon moldability was related to the monotonicity of chains of adjacent edges on its boundary, a result that we will also make use of. Given any two points p and q on ∂P, $[p \cdots q]$ represents all the points along the boundary ∂P between p and q in counterclockwise order, plus points p and q. (Note that points on the boundary ∂P can be any points on the boundary, not just vertices.) Given two points p and q on ∂P, $[p \cdots q]$ is a *monotone chain with respect to a line L* if the projection of all the points in $[p \cdots q]$ on L is in the same order as the points themselves. A polygon P is *2-monotone with respect to a line L* if ∂P can be partitioned into exactly 2 pieces that are both monotone chains with respect to L.



(a)

(b)

Figure 2. *(a) Polygon P (b) 2 monotone chains of ∂P with respect to L*

The relationship between monotonicity and moldability can be seen in Figure 2, where p and q are two points on ∂P that partition the boundary into two pieces $[p \cdots q]$ and $[q \cdots p]$, both of which are monotone chains with respect to the line L (thus the polygon is 2-monotone with respect to L). We can see that P is 2-moldable with opposite mold removal directions $\vec{d}$ and $-\vec{d}$

perpendicular to L, since $[p\cdots q]$ and $[q\cdots p]$ can be translated to infinity in directions $\vec{d}$ and $-\vec{d}$ respectively without collision with the interior of P. In fact, 2-moldability in opposite directions for a polygon P is equivalent to the 2-monotonicity of P with respect to some line L:

Theorem 1. *A polygon P is 2-moldable in opposite directions $\vec{d}$ and $-\vec{d}$ if and only if P is 2-monotone with respect to some line L with L perpendicular to $\vec{d}$.*

The proof, based on that presented in [9] for a more restrictive definition of a polygon and a looser definition of moldability, is presented in the appendix.

To find a parting direction, we take the approach of searching for a line L with respect to which the polygon is 2-monotone. We can determine whether such a line exists, and the orientation of all such lines, by examining a structure that represents the normals of the polygon's edges along with their connectivity. We call this structure the normal graph of the polygon. In the following section, we define the normal graph for a faceted polygon and show how it captures the directions of all lines L corresponding to (perpendicular) feasible parting directions. The key to an efficient algorithm, however, is to never actually build the entire normal graph, which would take time $O(n\log n)$. Instead, we build a summary structure in time $O(n)$ as described in section 5. Then in section 6 we describe how our algorithm is extended to handle the case of curved edges in the input.

3 The Normal Graph of a Faceted Polygon

The normal graph of a polygon is built on the unit circle. The unit normal of a (straight line) polygon edge is defined to be perpendicular to the edge and pointing away from the interior of the polygon. Translating the tail of these edge normal vectors to the origin places the heads on the unit circle; these points on the unit circle representing edge normal directions are called *normal points*. To build the normal graph, starting from any edge of the polygon, traversing the edges in counterclockwise order, we place the corresponding normal points on the unit circle. We connect each to the normal point of the next edge in the sequence with an oriented arc of the unit circle such that the angle of that arc is less than 180°. (We pre-process the input to merge and/or remove adjacent collinear edges with the same or opposite normals respectively, so that the arc angle will never be exactly equal to 180°.) These arcs connecting normal points are called *normal arcs*. When the last edge in the sequence around the polygon is reached, its normal point is connected to the normal point of the first edge in the sequence with an arc using the same criteria. The normal points and normal arcs together comprise the *normal graph*.

The visualization of the normal graph for an example polygon is shown in Figure 3. Since for a non-convex polygon the normal arcs will overlap each other, for visualization purposes we will always show a "tweaked" normal graph with the normal points and normal arcs expanded out from the unit circle so that the connectivity can be seen. The unit circle, as well as the normal vectors corresponding to each normal point, are shown in the center of the normal graph in gray. They are not considered part of the normal graph.

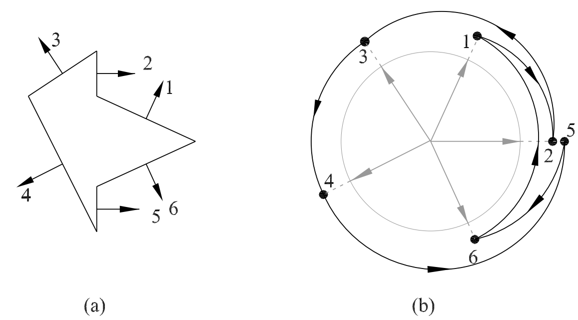


Figure 3. *(a) A six-sided polygon showing unit normals for each edge (b) Visualization of its normal graph*

We will test the 2-monotonicity of a polygon P with respect to some line L in order to determine the 2-moldability of P with opposite removal directions perpendicular to L. For a candidate monotone projection line L, we look at a corresponding line L' on the normal graph that passes through the origin of the unit circle and has the same orientation as L. L' is the corresponding *candidate partition line* (CPL) for the normal graph. If the polygon P is 2-monotone with respect to the line L, the corresponding CPL L' is called a *partition line* (PL) for the normal graph, since it partitions the normal graph into exactly two connected components, as we shall show.

If ∂P can be partitioned into two chains that are both monotone chains with respect to L, the edge normals for one chain will all point in directions to one side of (or along the direction of) L, and the edge normals for the other chain will all point in directions to the other side of (or along the direction of) L. Otherwise, there will be a collision of the mold when removed with the interior of the polygon, a contradiction with the 2-moldability of the polygon (since it is 2-monotone with respect to L and hence 2-moldable in opposite directions perpendicular to L). Thus in the normal graph, the corresponding normal points for one chain will lie on a closed semi-circle (i.e. a semi-circle including its endpoints) induced by the corresponding partition line L'. In Figure 4, for example, the edges 1, 2 and 3 on the polygon form a monotone chain with respect to the line L; thus the corresponding normal points 1, 2 and 3 lie on a closed semi-circle induced by the corresponding CPL L'.

Each of the normal points of the edges in a monotone chain can be connected to the normal point of the adjacent edge(s) in the monotone chain by an arc of the unit circle with an angle less than 180°. These arcs will also lie entirely on the same side of L', since all the normal points of the edges in a monotone chain lie on a semi-circle to one side of L'. These arcs are the normals arcs of the normal graph; hence any normal arcs that cross L'

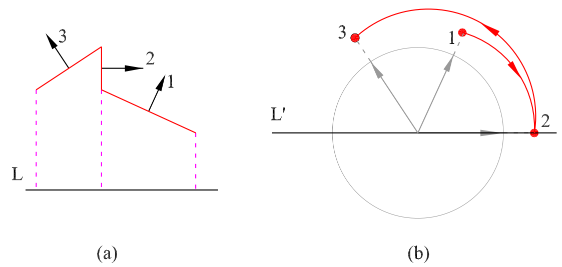


Figure 4. *(a) A monotone chain with respect to the line L (b) Normal arcs with angles less than* 180° *connecting its edges*

must connect edges in two different monotone chains. Thus a candidate partition line cuts the normal graph between normal points of edges in two different monotone chains once for each normal arc it intersects.

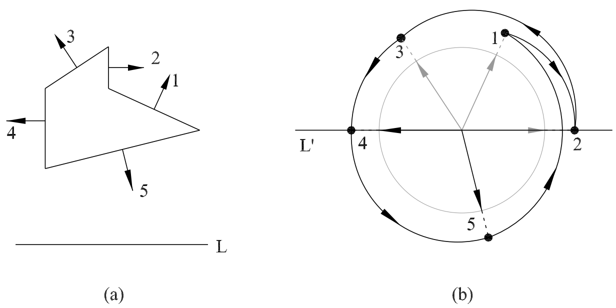


Figure 5. *A polygon and its corresponding normal graph with both inner points (3,4,5) and turning points (1,2) present*

Sometimes a normal point falls exactly on the CPL L' (this means the corresponding edge of the polygon is parallel to the parting direction $\vec{d}$ perpendicular to L). We distinguish two cases: *Turning points* are points where the two normal arcs they connect have opposite orientations. *Inner points* are points where the two normal arcs they connect have the same orientation. For a turning point on the CPL L' (see normal point 2 in Figure 5), the corresponding edge and the two adjacent edges must be in the same monotone chain with respect to the line L. For an inner point on the CPL (see normal point 4 in Figure 5), the two adjacent edges must be in two different monotone chains; the edge corresponding to the inner point can be put in either chain or divided between them anywhere along its length. Thus a candidate partition line cuts the normal graph once with each inner normal point it intersects, but does not cut the normal graph where it intersects a turning point.

Since the entire normal graph forms a connected loop, a candidate partition line L' that cuts it exactly twice will divide the corresponding polygon edges into exactly two monotone chains with respect to L (and hence correspond to an interference free parting direction for a two part mold with opposite removal directions).

Lemma 2. *A polygon P is 2-monotone with respect to a line L if and only if its normal graph can be partitioned into exactly two chains that are both composed of continuously connected arcs that lie on the two closed semi-circles on opposite sides of the corresponding CPL* L'.

Proof of Lemma 2

Proof. $\Longrightarrow$ If the polygon P is 2-monotone with respect to some line L, there must exist some partition of ∂P so that each of the two pieces is monotone with respect to some line L. For each monotone piece, all the normals of the edges must point away from L to the same side, which means that the normal points on the normal graph must lie on one closed semi-circle induced by the corresponding CPL L'. Since the edges on each polygon piece are continuously connected, the arcs connecting the corresponding normal points on the normal graph are all less than 180° and all in the same closed semi-circle.

$\Longleftarrow$ Suppose the normal graph can be partitioned into two chains by a CPL L'. We denote the normal points on one chain by $p_1, p_2, \cdots, p_n$, in counterclockwise order, and they represent the normals of edges $e_1, e_2, \cdots, e_n$. Since the chain of these points is composed of continuously connected arcs, we know that the edges $e_1, e_2, \cdots, e_n$ are continuously connected on the polygon P. Without loss of generality, we put the line L corresponding to L' horizontally, and all the normals of edges $e_1, e_2, \cdots, e_n$ point upward. From Figure 6, we can see that the projection of the edges $e_1, e_2, \cdots, e_n$ can only extend from right to left as long as the normals of the edges point upward, or the projection is a point on L if the edge is perpendicular to L. That means the projection of the points on the edges is in the same order as the points themselves. Therefore, the chain composed of $e_1, e_2, \cdots, e_n$ is a monotone chain with respect to L. Similarly, the remaining piece composed of the other edges on ∂P is also a monotone chain with respect to L. Thus by definition the polygon P is 2-monotone with respect to L.

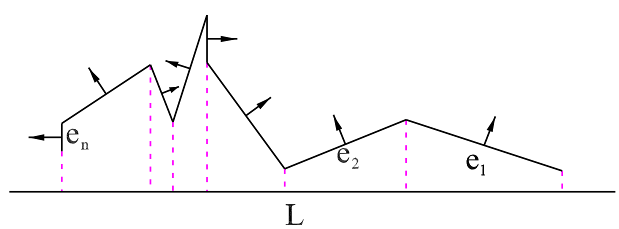


Figure 6. *The projection of line segment edges on a line L*

From Theorem 1 and Lemma 2, follows Theorem 3:

Theorem 3. *A polygon P is 2-moldable in opposite directions* $\vec{d}$ *and* $-\vec{d}$ *if and only if its normal graph can be partitioned into exactly two chains that are both composed of continuously connected arcs that lie on the two closed semi-circles on opposite sides of the CPL* L' *that is perpendicular to* $\vec{d}$.

Lemma 4. *There are exactly two intersections (excluding intersections at turning points) between a partition line L' and the normal graph, and these two points are separated on L' by the origin of the unit circle.*

Proof of Lemma 4:

Proof. Suppose the partition line L' exists. From Lemma 2, we know that the polygon P is 2-monotone with respect to some line L. Thus by definition there must exist some partition of ∂P such that each of the two polygon pieces is a monotone chain with respect to L.

Denote the two polygon pieces of ∂P by $[p\cdots q]$ and $[q\cdots p]$. Without loss of generality, we put L horizontally, the normal of each edge lying on the piece $[p\cdots q]$ pointing upward, and the normal of each edge lying on the piece $[q\cdots p]$ pointing downward. The point q may lie in the middle of an edge or at the intersection of two edges.

For the case where q lies in the middle of an edge, say edge 2, refer to Figure 7. In order to satisfy the monotonicity of $[p\cdots q]$ and $[q\cdots p]$, edge 2 must be perpendicular to L, and the adjacent edges 1 and 3 must lie on the right side of edge 2. The angles of the angles α and β are less than 180° on the normal graph, as is shown in Figure 7(b). Therefore, the connecting arcs must go from normal point 1 to normal point 2 and then to normal point 3. Since the two arcs are in the same orientation and normal point 2 lie on L', the normal graph crosses L' at the q end of $[p\cdots q]$.

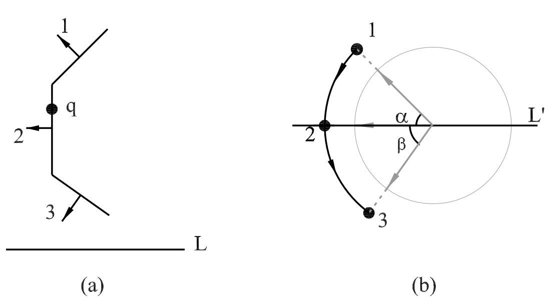


Figure 7. *Partition line intersects normal graph (case 1)*

For the case where the point q lies at the intersection of two edges, say edge S and T, refer to Figure 8. The angle γ is always less than 180°. Since the normal points representing S and T lie respectively above and below L' on the normal graph, the arc connecting them must cross L' at the q end of $[p\cdots q]$.

In a similar way, the normal graph must also cross L' once at the p end of $[q\cdots p]$. Since there are at most two intersections between L and the normal graph — otherwise the two chains separated by L' on the normal graph could not be continuously connected arcs, a contradiction with Lemma 2 — we reach the conclusion described in Lemma 4.

A simple case of a subset of polygon edges and a CPL which cannot be a PL is shown in Figure 9, in which the CPL L' passes

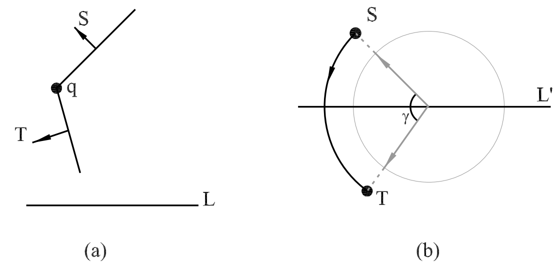


Figure 8. *Partition line intersects normal graph (case 2)*

through the normal arc connecting the normal points 1 and 2 and the normal arc connecting the normal points 2 and 3. This CPL partitions even the subset of the normal graph shown into three connected chains; thus it partitions the entire normal graph into more than two chains. Hence the polygon P containing the three edges shown is not 2-moldable in opposite directions perpendicular to L. Looking at Figure 9(a), we see that edge 2 points down away from L while edges 1 and 3 point up away from L; a collision with the interior of the polygon will occur when the upper half of the mold that shapes edges 1 and 3 is removed.

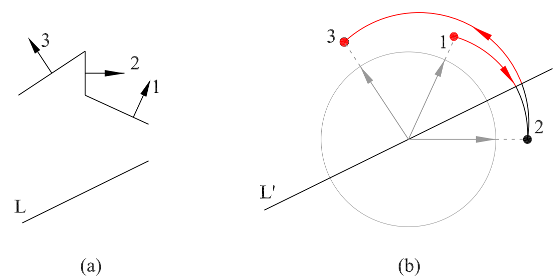


Figure 9. *(a) An edge chain with a line L (b) The CPL L' passing through two arcs that are connected by a turning point*

In Figure 10, on the other hand, the normal graph for a complete polygon containing the same segments is partitioned into exactly two connected chains by the CPL L'_1. Each chain consists of continuously connected arcs lying on one closed semicircle induced by L'_1, making L'_1 a partition line (PL). Therefore, the polygon P shown in Figure 10(a) is 2-moldable in opposite directions $\vec{d}_1$ and $-\vec{d}_1$ perpendicular to L'_1. However, the CPL L'_2 shown in Figure 11 is not a *PL* since the normal graph cannot be partitioned into exactly two chains by L'_2. That is, P is not 2-moldable in opposite directions perpendicular to L'_2.

4 Finding 2-Moldable Directions from the Normal Graph

In this section, we outline a procedure for finding valid parting directions from the normal graph. In the following section, we describe a similar procedure using a summary of the normal

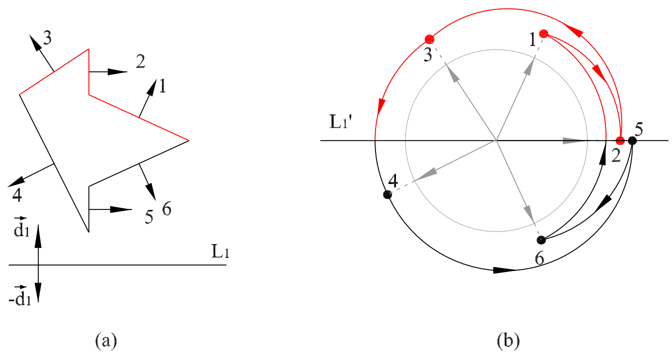


Figure 10. *(a) The polygon P (b) Normal graph of P with the CPL L_1'*

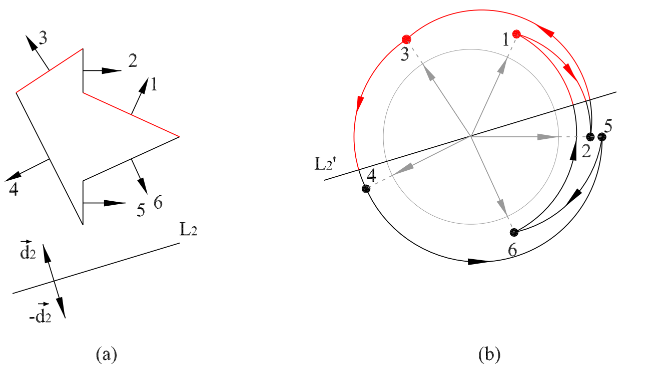


Figure 11. *(a) The polygon P (b) Normal graph of P with the CPL L_2'*

graph which can be built more efficiently than the full normal graph.

We can find all valid partition lines by radially sweeping (rotating) a CPL through 180°, determining at what angles (orientations) it intersects the normal graph at exactly two points (not including intersections with turning points). The valid parting directions are those perpendicular to these partition lines.

An example is illustrated in Figure 12. Imagine that the sweep begins with CPL orientation L_1' and continues counterclockwise through orientations L_2', L_3', ... $L_7' = L_1'$. The CPL L_1' intersects the normal graph at only two points; thus P is 2-moldable with removal directions perpendicular to L_1'. L_2' gives the same 2-moldability. Note that the CPL does not cross any turning points while rotating from L_1' to L_2'. The CPL L_3' goes through one turning point (point 2). It still gives the result of P being 2-moldable in directions perpendicular to L_3' since intersections between CPLs and the normal graph at turning points are not counted. After the sweep passes the turning point, the 2-moldability of P in directions perpendicular to the CPL changes. CPLs L_4' and L_5' each intersect the normal graph three times; thus P is not 2-moldable in directions perpendicular to them. When the sweep reaches the next turning point (point 1) at orientation L_6', the 2-moldability of P in directions perpendicular to the CPL changes again. When the sweep arrives at L_7', the line is back to the original L_1' orientation; the sweep stops here. As a result, we get all the 2-moldable directions, each of which is perpendicular to some CPL lying between L_1' and L_3' inclusive, or between L_6' and L_7' inclusive, in counterclockwise order.

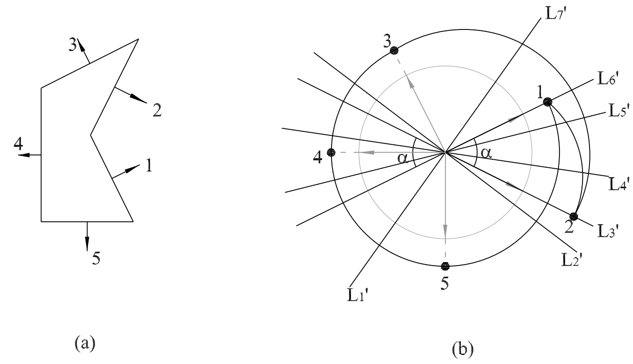


Figure 12. *Finding the 2-moldable directions for a polygon by sweeping CPLs*

Since the 2-moldability changes only when the rotating CPL crosses over one or more turning points, we don't need to test the infinite possible CPL orientations. We only need to test candidate orientations that pass through each turning point, and one representative orientation between each pair of turning points and/or their polar opposites that are radially adjacent (note that adjacent means not in the sense of connected in the normal graph, but adjacent in the order they are encountered by the sweep, which is equivalent to the ordering around the unit semi-circle of the turning points plus their polar opposites). If (and only if) the polygon is 2-moldable in the direction normal to this representative CPL, it will be 2-moldable for directions normal to all CPLs in this interval. This follows from the following lemma:

Lemma 5. *The result of the opposite direction 2-moldability test for a polygon P does not change unless the radial sweep of the candidate partition line crosses one or more turning points on the normal graph.*

Proof. Since the number of intersections between the candidate partition line and the normal graph does not change unless the sweep comes across one or more turning points on the normal graph, the 2-moldability in directions perpendicular to the CPL does not change.

Thus each candidate partition line passing through one or more turning points is checked for 2-moldability by counting its number of non-turning-point intersections with the normal graph. If there are more than two, P is not 2-moldable in directions perpendicular to the CPL, nor for directions perpendicular to orientations immediately clockwise or counter-clockwise from the

CPL. Otherwise, this CPL is a partition line, and CPLs immediately to one side or the other (but not both) may be as well. Candidate partition lines lying on the side on which the current turning point connects two arcs correspond to directions in which P is not 2-moldable; candidate partition lines lying on the other side correspond to directions in which P is 2-moldable unless another turning point has the same normal but connects two arcs on this other side. After testing all turning points in this manner, the results can be combined to give all the 2-moldable directions.

For the example shown in Figure 12, according to Lemma 5, we only need to check the CPLs going through each of the two turning points, one (turning point 1) passing through the normal point representing edge 1, and the other (turning point 2) passing through the normal point representing edge 2. The polygon P is 2-moldable in the directions perpendicular to either of these two CPLs. For turning point 1, CPLs rotated immediately clockwise from L_6' (interior to angle α) are not partition lines, and the CPLs rotated immediate counter-clockwise (exterior to α) are partition lines. Similarly for turning point 2, with the directions of rotation reversed. Combining these results, we find that the polygon P is 2-moldable in directions that are perpendicular to any CPL lying outside the open intervals defined by double-angle α.

5 Summary Normal Graphs for Improved Efficiency

The radial sweep procedure outlined above relies on first building the full normal graph, which has running time $O(n \log n)$ because of the necessity of sorting the normal points into radial order. Our more efficient $O(n)$ algorithm exploits the fact that we do not need all of the information contained in the normal graph during the sweep: we only need to know whether or not a candidate partition line intersects the normal graph *more than* two times, not the *exact* number of times it intersects.

As we saw above, we can ignore non-turning points during the sweep, so there is no reason we need to keep track of them in our normal graph summary structure. We collapse normal arcs separated by non-turning points into composite normal arcs. Furthermore, we can ignore turning points that fall within an interval that we have already determined corresponds to non-feasible directions. For example, for the geometry shown in Figure 13(a), once we have built up the partial normal graph from segments 1, 2, and 3, we know that there are no feasible partition lines that intersect the clockwise interval between points 2 and 3 in the normal graph; we can thus omit points 4 and 5 from the summary graph since they give us no new information about moldability. Our summary graph only includes turning points corresponding to potentially feasible directions, those where there are fewer than two composite normal arcs spanning/crossing the turning point, and records whether there are zero or one normal arcs spanning it. Between radially adjacent members of this subset of turning points, we only keep track of whether there are zero, one, or multiple composite normal arcs spanning that interval, not the exact number of arcs or what points in the full normal graph the arcs would connect. The final summary graph for polygon 12(a) is shown in Figure 13(c).

We store our summary normal graph in a doubly-linked circular list data structure, where each node corresponds to a turning point and the next and previous links point to the next clockwise and counterclockwise turning points in the normal graph summary we have built up thus far. For each node, in addition to storing the normal direction and a count of whether there are zero or one composite normal arcs spanning the node, we also store a count of whether there are currently zero, one, or multiple composite normal arcs spanning the interval between this node and its current clockwise and counterclockwise neighbors in the summary graph. This number of spanning arcs for each node and each link is called the node or link's count. We initialize this circular linked list structure with the first two turning points, forming a simple two-node doubly linked list. (If there were *no* turning points, all directions are feasible and we are done.) The node counts are both zero; the link corresponding to the normal arc(s) between the turning points has count one and the other link has count zero. An example of an initial two-node summary graph is shown in Figure 14(a) for the polygon from Figure 13.

We use an incremental approach to building the rest of the summary normal graph. For each subsequent turning point processed, we follow the links from the previous turning point in either the clockwise or counterclockwise direction (depending on the orientation of the normal arcs in between the previous and current turning points) to find the correct position for the new turning point. As we traverse each link and node, we increment its count (as well as the count of the corresponding link in the opposite direction) to reflect the composite normal arc between the last and current turning point. If this would cause a node count to exceed one, we delete that node from the linked list. A link count of "multiple" just remains the same when incremented. When we find the correct position for the new turning point, we insert the new node, setting its count equal to that of the links that connected its two neighbors. If the new turning point is to be inserted between two nodes that already have a link count greater than one, we do not insert it. Similarly, if there is already a node with the same normal, we just increment the link count. We start the position search for the next turning point from the node we just inserted (or else from the corresponding existing node, or else from the node that would have been its successor had we inserted it). The incremental state of the summary normal graph at each stage of building the polygon from Figure 13 is shown in Figure 14.

To find the next turning point, we process the input normals sequentially in their counterclockwise order around the boundary of the input polygon. We calculate the direction of, and angle subtended by, the normal arcs between each normal and the previous normal processed. If the direction changed between clockwise and counterclockwise, the normal is a turning point. We then add up all the subtended angles since the last turning point to find the angle of the composite normal arc between the two turning points, before adding the new turning point to the summary graph. If this total angle exceeds $540°$, or one and a half

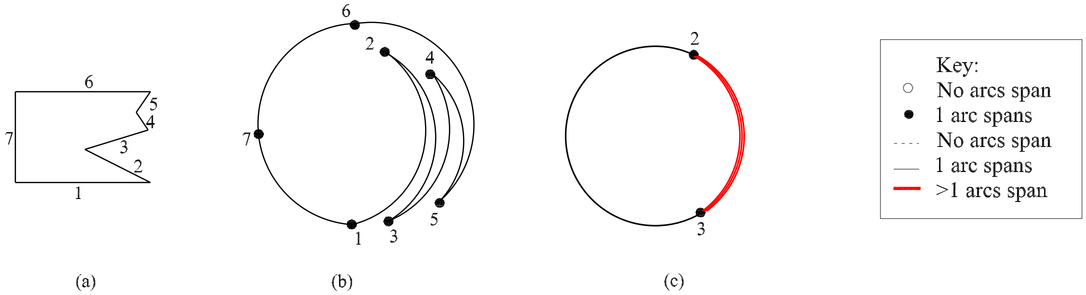


Figure 13. *(a) A polygon (b) Its normal graph (c) Its summary normal graph*

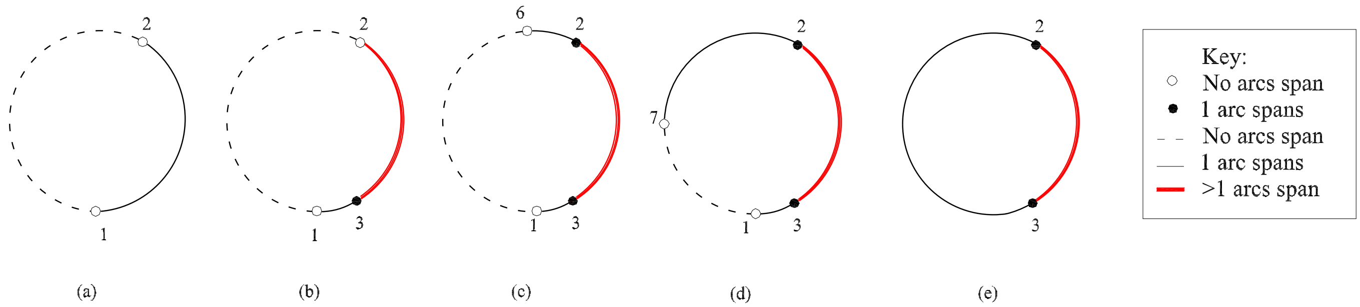


Figure 14. *Building the summary graph: (a) state after normals 1 and 2 are processed; (b) state after normals 1-3 are processed, which remains unchanged after normals 4 and 5 are processed; (c) state after normals 1-6 are processed; (d) state after normals 1-7 are processed; (e) final summary graph. (For clarity, this figure shows the state after each normal is processed, but in practice our implementation folds updates of non-turning normals into the processing of turning normals.)*

times around the normal graph, then there are no feasible mold removal directions and we exit the program. Otherwise, if the total angle exceeds 360°, before inserting the new node we first make an entire traversal once around the linked list incrementing the count of each node and link (and deleting nodes whose counts exceed one). Then we use the standard procedure (traversal plus count increment and possible node deletion) to find the position for and insert the new turning point.

This algorithm takes time $O(n)$. There are a maximum of n nodes to be added to the circular linked list. Although the insertion of any individual node may require traversing all the way around the linked list (possibly twice in the case of a 540° composite normal arc), we can traverse from one side of each node to the other only once before the next traversal causes it to be deleted (since each traversal updates the node's count and nodes are deleted when their count exceeds one). Thus the total number of traversals over all nodes will be bounded by $O(n)$. Crossing over and updating a node and its links takes constant time; deleting a node takes constant time; and inserting a node once its position is found takes constant time. Thus the entire process of building the summary normal graph takes time $O(n)$.

To find the intervals corresponding to feasible removal directions, we again use a radial sweep of a CPL, this time through the summary graph. The feasible directions are perpendicular to those intervals where the sum of the counts for the nodes and/or arc links intersected by the two sides of the CPL is two. For the CPL through the first turning point we process in the completed summary graph, we must follow the links around the graph to find the opposite link in order to calculate the total number of intersections for this CPL, taking up to $O(n)$ time for the first test. We then track two pointers, one at the first turning point's link and one at its opposite link that we've just located. We increment these pointers in tandem, advancing both in the same direction, alternating between them so that we always increment the pointer with the shortest distance to the next turning point. This allows us to check the CPLs associated with each remaining turning point in constant time, without requiring an $O(n)$ traversal to find the opposite link. Thus finding all feasible directions from the summary graph also takes $O(n)$, for a total running time of $O(n)$.

6 Finding 2-moldable Directions for a Polygon with Curved Edges

Next, we show how our algorithm can be applied to the case of a polygon with curved edges. The normal graph of such a polygon will contain a continuous range of contiguous normal points corresponding to each curved edge. Turning points can occur only at the endpoints of each such "normal range" if we first break each curved edge up into *simple curves*:

Definition 1. *A* simple curve *is a G^1-continuous curve without any inflection points.*

It follows from the definition that the tangent directions of a simple curve change continuously and monotonically (toward one direction, clockwise or counterclockwise) when moving along the directed edge from one endpoint to the other; thus the normal directions also change continuously and monotonically.

The curvature at each point on a simple curve has the same sign, i.e., positive or negative. (We follow the standard convention that the curvature is positive if the center of curvature is on the left when moving along a curve and negative if the center of curvature is on the right [11], as shown in Figure 15.)

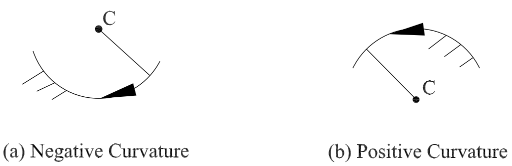


Figure 15. *Sign of curvature for directed edges (the cross-hatched area denotes the interior of the polygon).*

We could imagine decomposing each simple curve into n approximating straight line segments. For each approximating line segment, there is a normal point representing its normal direction. An example of building the normal graph of a polygon whose curved edge is approximated by seven straight line edges is shown in Figure 16. Since the normal directions change toward one direction along the simple curve, the arcs connecting the normal points that represent the approximating line segments all have the same orientation. Thus only the first and the last among these normal points could potentially be turning points, even as n goes to infinity. Therefore, the whole range of normal points can be simplified to a single arc on the normal graph whose endpoints are the normals perpendicular to the tangent directions at the endpoints of the simple curve. The orientation of the arc depends on the sign of the curvature of the simple curve: counterclockwise if the curvature is positive, clockwise if the curvature is negative.

Definition 2. *The* beginning/ending normal directions of a simple curve *are the normal directions, pointing away from the interior of the polygon, of that curve at its beginning/ending points respectively.*

Let p and q denote the endpoints of a simple curve. Then no normal point on the normal graph representing the normal direction at a point in the open interval $(p \cdots q)$ could be a turning point, since the normal direction changes along the curve in only one direction (clockwise or counterclockwise). They are all inner points on the normal graph. From Lemma 5, we know that the inner points won't affect the polygon's 2-moldability. So only the candidate partition lines passing through the two endpoints of each simple curve may need to be checked for 2-moldability (if these endpoints are also the turning points on the normal graph).

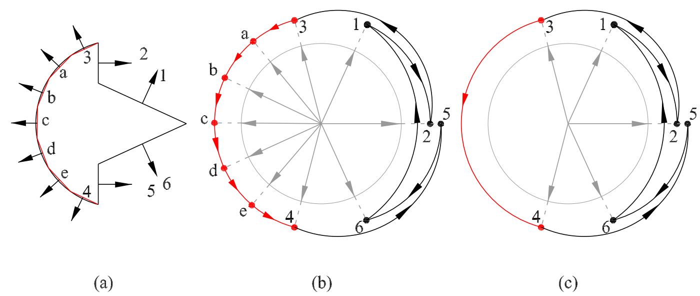


Figure 16. *(a) A polygon with a curved edge where the curve is approximated by 7 straight line segment edges, labeled* $3, a, b, c, d, e,$ *and* 4. *(b) The corresponding normal graph. (c) The arcs connecting the normal points for these edges can be simplified to a single oriented arc.*

Thus for input containing arbitrary curved edges, we first decompose each curve into simple curves by breaking it at its inflection points, or any point that interrupts the G^1-continuity of the curve. The curves are broken at these points because the normal direction changes there, interrupting the continuity of the normal directions along the curve. The decomposition is illustrated in Figure 17.

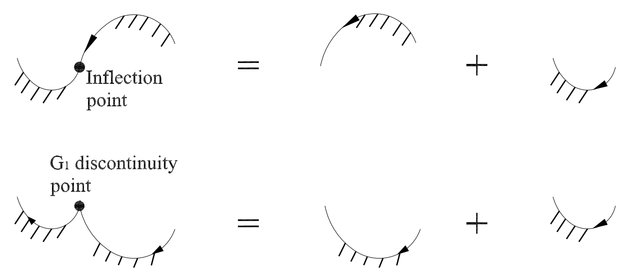


Figure 17. *The decomposition of general curved edges into simple curves. The hatched area denotes the interior of the polygon.*

We do need to consider a special case for curved edges that we cannot eliminate via preprocessing as we could for the straight line case. For faceted polygons, the case of two adjacent boundary segments with opposite normals could be ignored by removing the overlapping length of the segments that defined zero-area geometry. But with curved polygons, the starting normal of one curve segment may legitimately be exactly opposite the ending normal of the previous curve (or straight line) segment, as shown in Figure 18(a). In this case, we must disambiguate between whether the normal arc that connects them in the normal graph should be clockwise or counter-clockwise, since both choices for normal arcs subtend exactly 180°. We can imagine tweaking the endpoints of both segments by ε outwards along their normals in order to insert between them an ε-radius semicircular curved segment, tangent to both tweaked points, with direction matching those of the tweaked segments (see Figure 18(c)).

As ε approaches zero, in the limit we have our original geometry. The sign of the curvature of this inserted semicircular curve segment tells us whether to connect the original tweaked curves using a clockwise or counterclockwise normal arc.

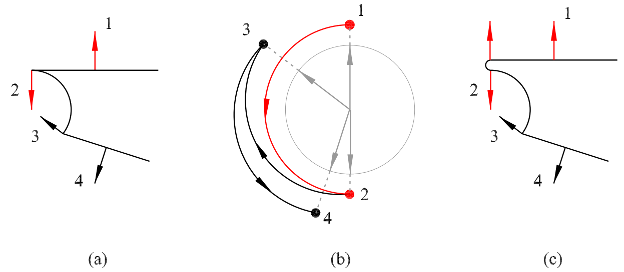


Figure 18. *The normal graph with the angle of an arc equal to* 180°

Thus for input with curved edges, our full procedure is as follows. First, break curved input edges into simple curves. Next, calculate the sign of curvature and the starting and ending normal for each simple curve. Third, insert infinitesimal semicircular dummy edges with appropriate positive or negative curvature between adjacent edges whose normal directions where they meet differ by exactly 180°. Fourth, process the list of normals and curvature signs, building the summary normal graph from the turning points as above, with the variation that the sign of the curvature is used to orient normal arcs between the start and end normals of the same curve segment. The same rule as before (the direction such that the angle of the arc is less than 180°) is used to orient normal arcs connecting normal points on adjacent but separate boundary segments. Finally, use the radial sweep to output the feasible directions indicated by the summary normal graph.

7 Conclusions

We have presented a new $O(n)$ algorithm for determining all feasible parting directions for a 2D polygon to be molded in a two part, rigid, reusable mold with opposite removal directions. We have designed the algorithm so that it can handle both faceted polygons and those with curved edges using an identical normal graph summary structure. We find *all* feasible removal directions, even though only one direction will ultimately be chosen for molding, because multiple choices allow us to optimize with respect to additional manufacturing considerations such as mold flow, shrinkage, parting plane geometry, gate/runner placement, etc.

Of course, ultimately we want to determine moldability for 3D geometries. In future work, we will study how the analysis of 2D input contours defining 3D CAD extrusion, sweep, and loft operations, combined with Boolean operations, can be used for incremental moldability analysis of 3D geometries in real time while the user designs them.

ACKNOWLEDGMENTS

This research was supported in part by UC MICRO 02-068 and 03-077, the UCB College of Engineering, and a UCB Committee on Research Junior Faculty Research Grant. We also thank the reviewers for helpful suggestions.

REFERENCES

[1] Wong, T., Tan, S. T., and Sze, W. S., 1998. "Parting line formation by slicing a 3D CAD model". Engineering with Computers, **14** (4) , pp. 330–343.

[2] Ravi, B., and Srinivasan, M. N., 1990. "Decision criteria for computer-aided parting surface design". *Computer-Aided Design,* **22** , pp. 11–18.

[3] Chen, Y. H., 1997. "Determining parting direction based on minimum bounding box and fuzzy logics". Int. J. Mach. Tools Manufact., **37** (9) , pp. 1189–1199.

[4] Hui, K. C., and Tan, S. T., 1992. "Mould design with sweep operations - a heuristic search approach". Computer-Aided Design, **24** (2) February , pp. 81–91.

[5] Hui, K. C., 1997. "Geometric aspects of the mouldability of parts". Computer-Aided Design, **29** (3) , pp. 197–208.

[6] Dhaliwal, S., Gupta, S., Huang, J., and Kumar, M., 2001. "A feature based approach to automated design of multi-piece sacrificial molds". ASME Journal of Computing and Information Science in Engineering, **1** (3) , pp. 225–234.

[7] Chen, Y., 2001. *Computer-Aided Design for Rapid Tooling: Methods for Mold Design and Design-for-Manufacture.* PhD thesis, Georgia Institute of Technology.

[8] Stampfl, J., Liu, H.-C., Nam, S. W., Sakamoto, K., Tsuru, H., Kang, S., Cooper, A. G., Nickel, A., and Prinz, F. B., 2002. "Rapid prototyping and manufacturing by gelcasting of metallic and ceramic slurries". Materials Science & Engineering., **334** (1-2) Sep , pp. 187–192.

[9] Rappaport, D., and Rosenbloom, A., 1994. "Moldable and castable polygons". *Computational Geometry,* **4** , pp. 219–233.

[10] Ahn, H.-K., de Berg, M., Bose, P., Cheng, S.-W., Halperin, D., Matousek, J., and Schwarzkopf, O., 2002. "Separating an object from its cast". *Computer-Aided Design,* **34** , pp. 547–59.

[11] Gray, A., 1993. *Modern Differential Geometry of Curves and Surfaces.* CRC Press, Baca Raton, Florida.

A Additional Proofs

Proof of Theorem 1:

Proof. Rappaport and Rosenbloom showed that if $[p\cdots q]$ is removable in direction $\vec{d}$, then $[p\cdots q]$ is monotone with respect to a line L perpendicular to $\vec{d}$ [9]. We also know by definition that if a polygon P is 2-moldable in opposite directions $\vec{d}$ and $-\vec{d}$, then ∂P can be partitioned into exactly 2 pieces, say $[p\cdots q]$ and $[q\cdots p]$, that can be translated to infinity in directions $\vec{d}$ and $-\vec{d}$ respectively without collision with the interior of P. In other words, $[p\cdots q]$ and $[q\cdots p]$ are removable in directions $\vec{d}$ and $-\vec{d}$ respectively. Therefore, $[p\cdots q]$ and $[q\cdots p]$ are both monotone with respect to a line L perpendicular to $\vec{d}$. Thus by definition the polygon P is 2-monotone with respect to the line L.

In the same literature, it was also proved that if $[p\cdots q]$ is monotone with respect to a line L_p, and $[q\cdots p]$ is monotone with respect to a line L_q, then there are directions $\vec{d}_{pq}$ and $\vec{d}_{qp}$ perpendicular to L_p and L_q respectively such that $[p\cdots q]$ is removable in direction $\vec{d}_{pq}$ and $[q\cdots p]$ is removable in direction $\vec{d}_{qp}$. If L_p and L_q are the same line and denoted by L, our case becomes a special case in that $[p\cdots q]$ and $[q\cdots p]$ are monotone with respect to the same line L and hence the polygon P is 2-monotone with respect to L, by definition. If we use $\vec{d}$ for $\vec{d}_{pq}$, $\vec{d}_{qp}$ can be denoted by $-\vec{d}$, and we can rephrase the conclusion by stating that $[p\cdots q]$ and $[q\cdots p]$ are removable in opposite directions $\vec{d}$ and $-\vec{d}$ respectively. That is, $[p\cdots q]$ and $[q\cdots p]$ can be translated to infinity in directions $\vec{d}$ and $-\vec{d}$ respectively without collision with the interior of P.