Partitioning a Polygon into Two Mirror Congruent Pieces

Dania El-Khechen* Thomas Fevens* John Iacono† Günter Rote‡

1 Introduction

Polygon decomposition problems are well studied in the literature [6], yet many variants of these problems remain open. In this paper, we are interested in partitioning a polygon into mirror congruent pieces. Symmetry detection algorithms solve problems of the same flavor by detecting all kinds of isometries in a polygon, a set of points, a set of line segments and some classes of polyhedra [2]. Two open problems with unknown complexity were posed in [2]: the minimum symmetric decomposition (MSD) problem and the minimal symmetric partition (MSP) problem. Given a set D in R^d ($d \in 2, 3$), the goal is to find a set of symmetric (nondisjoint for MSD and disjoint for MSP) subsets $\{D_1, D_2, \ldots, D_k\}$ of D such that the union of the D_i is D and k is minimum. The following problem is a decision version of MSP where $k = 2$:

Problem 1 *Given a polygon P with n vertices, compute a partition of P into two (properly or mirror) congruent polygons P_1 and P_2, or indicate such a partition does not exist.*

Erikson claims to solve the aforementioned problem in $O(n^3)$ [4]. Rote observes that a careful analysis of Erikson's algorithm yields a $O(n^3 \log n)$ running time for proper congruence and he shows that the combinatorial complexity of an explicit representation of the solution in the case of mirror congruence cannot be bounded as a function of n [7]. Rote also gives a counterexample where the algorithm fails for a polygon with holes. An $O(n^2 \log n)$ algorithm to solve the problem for properly congruent and possibly nonsimple P_1 and P_2 was presented recently [3]. It was also conjectured that the output can be restricted to simple polygons without an increase in the runtime [3]. In this paper, we present an $O(n^3)$ algorithm to solve the problem for mirror congruent and possibly nonsimple polygons P_1 and P_2. In other words, our algorithm is able to produce solutions unbounded by n in a time polynomial in n using an implicit representation of the output. Note that we can restrict the output to simple polygons if we allow an additional linear factor for intersection checking.

*Department of Conputer Science and Software Engineering, Concordia University, Montréal, Québec, Canada.

†Department of Computer and Information Science, Polytechnic Institute of New York University, 5 Metrotech Center, Brooklyn NY 11201 USA. http://john.poly.edu. Research partially supported by NSF Grants CCF-0430849 and OISE-0334653 and by an Alfred P. Sloan Research Fellowship. Research partially completed while the author was on sabbatical at the School of Computer Science, Mcgill University, Montréal, Québec, Canada.

‡Freie Universität Berlin, Institute of Computer Science, Germany

2 Preliminaries

Two polygons are *mirror congruent* (*properly congruent*) if they are equivalent up to reflection or glide reflection (rotations and translations). Note that a glide reflection is a reflection followed by a translation parallel to the reflection axis. A reflection along an axis g followed by a rotation or a translation is a reflection around an axis g'. In this paper, we focus on mirror congruent polygons. Congruence transforms involving glide reflection are denoted by $T = (g, \mathbf{v})$ where g is the axis of reflection and $\mathbf{v}$ is the vector of translation if any. Let $T^{-1} = (g, -\mathbf{v})$. We refer to the boundary of a polygon P by $\delta(P)$ and we normalize P to have unit perimeter. A polyline that is a subset of $\delta(P)$ is specified by a start point and an endpoint on $\delta(P)$ (not necessarily vertices) and is always considered to be directed clockwise around P. A polyline can be viewed as an alternating sequence of lengths and angles, which always begins and ends with a length. Two polylines are congruent if they are represented by the same sequence, two polylines are *flip congruent* if they are represented by the same sequence after replacing all of the angles α_i in one by $2\pi - \alpha_i$ and reversing the order of the sequence, and two polylines are *mirror congruent* if they are represented by the same sequence after reversing the order of the sequence. Let $\underset{P}{\angle} a$ be the interior angle of a point a on a polygon P. Let $\overline{ab}$ be the line segment with endpoints a and b and $P[a \mathinner{..} b]$ be the polyline connecting a to b on P in clockwise order. We use $\overset{\text{FLIP}}{\cong}$ to denote flip congruence, $\overset{\text{MIRROR}}{\cong}$ to denote mirror congruence. Observe that $P[a \mathinner{..} b] \overset{\text{FLIP}}{\cong} P[b \mathinner{..} a]$. Let $vd(a, b)$ be the vertical distance between the two points a and b. A partitioning of P, if it exists, is a solution to Problem 1 and is denoted by $S = (P_1, P_2)$. It consists of polygons P_1 and P_2 such that there exists a transformation where $T_S(P_1) = P_2$. The *split polyline*, denoted by Split(S), partitions the polygon P into P_1 and P_2. We are interested in a split polyline that has minimum complexity. When P is symmetric, we call the partition trivial and the problem reduces to symmetry detection which has been solved in linear time in [2]. Note if T_S is a reflection it can be determined by one pair of points $(p_i, T_S(p_i))$ such that $p_i \in \delta(P_1)$ and $T_S(p_i) \in \delta(P_2)$. If T_S is glide reflection, it can be determined by two pairs of points $(p_i, T_S(p_i))$ and $(p_j, T_S(p_j))$ such that p_i and p_j belong to $\delta(P_1)$ and $T_S(p_i)$ and $T_S(p_j)$ belong to $\delta(P_2)$. We say that two subsets $s_1 \subseteq P_1$ and $s_2 \subseteq P_2$

of congruent polygons P_1 and P_2 are transformationally congruent with respect to congruence transformation T_S if $T_S(s_1) = s_2$.

3 Results

3.1 Preprocessing

Congruence of polylines is detected by string matching. Our string representation of polygons and polylines yields Lemma 2.

Lemma 2 ([5]) *Given a polygon P, with $O(n^2)$ preprocessing and space, queries of the form $P[a\ ..\ b] \overset{?}{\overset{\text{MIRROR}}{\cong}} P[c\ ..\ d]$ and $P[a\ ..\ b] \overset{?}{\overset{\text{FLIP}}{\cong}} P[c\ ..\ d]$ can be answered in constant time.*

Let the length of a polyline $P[a\ ..\ b]$ (denoted $d_P(a,b)$) be the sum of the lengths of all the segments that form this polyline. Let $d_P^{-1}(a,x)$ be the point b such that $d_P(a,b) = x$. That is, it is the point on $\delta(P)$ obtained by walking x units clockwise around $\delta(P)$ from a. Note that $d_P^{-1}(a,0.5) = b$ is equivalent to $d_P^{-1}(b,0.5) = a$.

Lemma 3 ([1]) *Given a polygon P, with $O(n)$ preprocessing and space, the functions d_P and d_P^{-1} can be computed in constant time if the endpoints are vertices of the given polygon, and in $O(\log n)$ if they are not, using standard point location techniques.*

3.2 Algorithms

Lemma 4 *Assume that P can be nontrivially partitioned into two mirror congruent polygons where $S = (P_1, P_2)$ and let b and e denote the endpoints of the split polyline* Split(S) *then either $P_1[b\ ..\ e]$ is disjoint from the polyline $T_S\,(P_1[b\ ..\ e])$, $T_S\,(P_1[b\ ..\ e])$ partially overlaps with $P_1[b\ ..\ e]$, or $P_1[b\ ..\ e]$ and $P_2[e\ ..\ b]$ are line segments.*

Proof. Suppose that $T_S\,(P_1[b\ ..\ e]) = P_2[e\ ..\ b]$. We know that by definition $P_1[b\ ..\ e] \overset{\text{FLIP}}{\cong} P_2[e\ ..\ b]$. Therefore, the polyline $P_1[b\ ..\ e]$ and its flip congruent $P_2[e\ ..\ b]$ are mirror congruent which obviously cannot happen unless $P_1[b\ ..\ e]$ and $P_2[e\ ..\ b]$ are line segments. □

In section 3.3, we present an algorithm for the case where Split(S) is disjoint from $T_S\,$(Split(S)) (see Figure 1) and in section 3.4, we present an algorithm for the case where they partially overlap (see Figure 2). All the proofs in the following sections are omitted due to space constraints.

3.3 Disjoint split polyline

In this section, we assume that if a solution exists then the split polyline Split(S) is disjoint from its mirror image by the transformation T_S. We first show the necessary conditions for the existence of a solution in Lemma 5, namely that a solution $S = (P_1, P_2)$ can be specified by a six-tuple of points on $\delta(P)$ satisfying some properties. In Lemma 6, we show how to verify if a given six-tuple specifies a valid solution or not. In Lemmas 7 and 8, we show how, given two points of a solution six-tuple, we can find the rest of the points in the six-tuple. In Theorem 1, given that (by Lemma 5) at least four points of a solution six-tuple are vertices, we present an $O(n^3)$ algorithm that solves Problem 1 for the case discussed in this section.

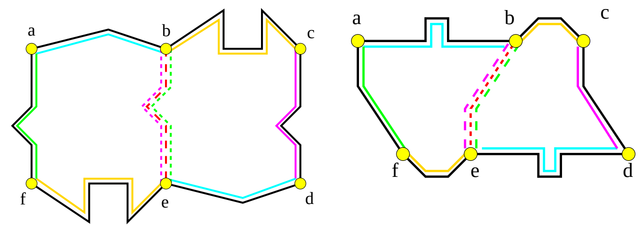

Figure 1: Polygons partitioned into two simple mirror congruent pieces with a nonoverlapping split polyline

For Lemmas 5, 6, 7 and 8, assume that P can be nontrivially partitioned into two mirror congruent polygons P_1 and P_2 where $S = (P_1, P_2)$ and Split(S) is disjoint from $T_S\,$(Split(S)). Let $d = T_S(b)$, $c = T_S(e)$, $f = T_S^{-1}(b)$, and $a = T_S^{-1}(e)$.

Lemma 5 *The following facts hold (see Figure 1): (a,b,c,d,e,f) appear in clockwise order on $\delta(P)$; $P[f\ ..\ a] \overset{\text{MIRROR}}{\cong} P_2[e\ ..\ b]$; $P[c\ ..\ d] \overset{\text{MIRROR}}{\cong} P_1[b\ ..\ e]$; $P[a\ ..\ b] \overset{\text{MIRROR}}{\cong} P[d\ ..\ e]$; $P[b\ ..\ c] \overset{\text{MIRROR}}{\cong} P[e\ ..\ f]$; $P[f\ ..\ a] \overset{\text{FLIP}}{\cong} P[c\ ..\ d]$; $\underset{P}{\angle} a + \underset{P}{\angle} c = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$; $\underset{P}{\angle} f + \underset{P}{\angle} d = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$; at least two of the points in $\{a,c,e\}$ and two of the points in $\{b,d,f\}$ are vertices of P; $d_P^{-1}(a,0.5) = d$; $d_P^{-1}(b,0.5) = e$; and $d_P^{-1}(c,0.5) = f$.*

Lemma 6 *Given the preprocessing in Lemma 2 and the positions of six points (a,b,c,d,e,f) on $\delta(P)$, it can be checked that the points specify a valid solution $S = (P_1, P_2)$ for the disjoint split polyline case of Problem 1 in constant time.*

Lemma 7 *The points (a,b,c,d,e,f) are as defined in Lemma 5. Given the position of two points of $\{a,c,e\}$ or $\{b,d,f\}$ and the preprocessing in Lemma 3, the positions of all six points (a,b,c,d,e,f) can be computed $O(\log n)$ time except in the case where both b and e are not vertices of P.*

Lemma 8 *Given the positions of $\{a,c,d,f\}$, the fact that both b and e are not vertices (equivalent to $\{a,c,d,f\}$ being all vertices by Lemma 5) and the preprocessing in Lemma 2, the positions of b and e can be computed $O(n)$ time.*

Theorem 1 *Given a simple polygon P and given that* Split(S)*, if it exists, is disjoint from $T_S\,(\text{Split(S)})$, a solution $S = (P_1, P_2)$ to Problem 1 can be found in $O(n^3)$ time if and only if P can be partitioned into two congruent polygons.*

3.4 Partially overlapping split polyline

In this section, we assume that if a solution S exists then the split polyline Split(S) is partially overlapping with its mirror image by the transformation T_S. We first show the necessary conditions for the existence of a solution in Lemma 9, namely that a solution $S = (P_1, P_2)$ can be specified by a six-tuple of points on $\delta(P)$ that obey one of two sets of properties (which we call case 1 and case 2). In Lemma 10, we show how to verify if a given six-tuple specifies a valid solution or not. In Lemmas 11 and 12, we show how, in each one of the two cases, given two points of a solution six-tuple, we can find the rest of the six-tuple points. In Theorem 2, given that (by Lemma 9) at least four points of a solution six-tuple are vertices, we present an $O(n^3)$ algorithm that solves Problem 1 for the case discussed in this section.

For Lemmas 9, 10, 11 and 12, assume that P can be nontrivially partitioned into two mirror congruent polygons P_1 and P_2 where Split(S) is partially overlapping with $T_S\,(\text{Split(S)})$. Let $T_S(e) = c$, $T_S^{-1}(b) = f$. Assume without loss of generality that the axis of glide reflection g is vertical.

Lemma 9 *The following facts hold (see Figure 2): $P[e \,..\, f] \overset{MIRROR}{\cong} P[b \,..\, c]$; $P_1[f \,..\, e] \overset{MIRROR}{\cong} P_2[c \,..\, b]$; there exists two points a and d on $\delta(P)$ such that either $P[f \,..\, a] \overset{MIRROR}{\cong} P[c \,..\, d]$, $P[a \,..\, b] \overset{FLIP}{\cong} P[d \,..\, e]$, $\underset{P}{\angle}(2\pi - \underset{P}{\angle} d) + \underset{P}{\angle} f = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$ and $\underset{P}{\angle} c + \underset{P}{\angle}(2\pi - \underset{P}{\angle} a) = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$ (this is case 1, see the left polygon in Figure 2) or $P[f \,..\, a] \overset{FLIP}{\cong} P[c \,..\, d]$, $P[a \,..\, b] \overset{MIRROR}{\cong} P[d \,..\, e]$, $\underset{P}{\angle} d + \underset{P}{\angle} f = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$ and $\underset{P}{\angle} c + \underset{P}{\angle} a = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$ (this is case 2 see the right polygon in Figure 2); at least two of the points in $\{a, c, e\}$ and two of the points in $\{b, d, f\}$ are vertices of P; if $x = (vd(b, e)/vd(f, b)) \bmod vd(c, d)$ then for case 1, x is an odd number and for case 2, x is even; (a, b, c, d, e, f) appear in clockwise order on $\delta(P)$; $d_P^{-1}(a, 0.5) = d$; $d_P^{-1}(b, 0.5) = e$ and $d_P^{-1}(c, 0.5) = f$.*

Lemma 10 *Given the preprocessing in Lemma 2 and the positions of six points (a, b, c, d, e, f) on $\delta(P)$, it can be checked that the points specify a valid solution $S = (P_1, P_2)$ for the partially overlapping split polyline case of Problem 1 in constant time.*

Lemma 11 *The points (a, b, c, d, e, f) are as defined in Lemma 9. Given the position of any two of $\{a, c, e\}$ or $\{b, d, f\}$ and the preprocessing in Lemma 3, the positions of all six points (a, b, c, d, e, f) can be computed in $O(\log n)$ time except in the cases where either both b and e or both c and f are not vertices (Figures 3 and 4).*

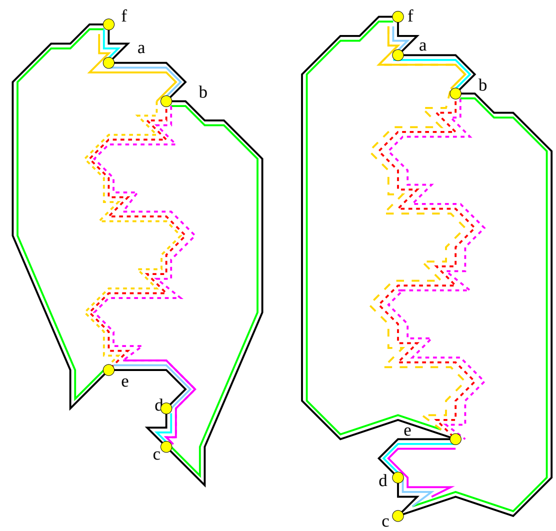

Figure 2: Polygons partitioned into two simple mirror congruent pieces with an overlapping split polyline.

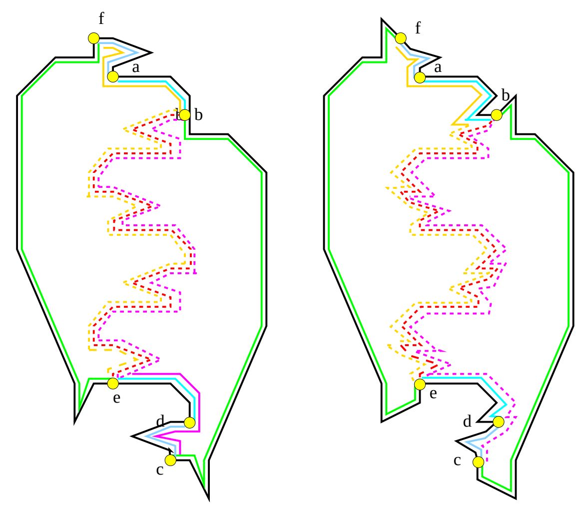

Figure 3: Case 1a (left) where $\{a, c, d, f\}$ are vertices and $\{b, e\}$ are not. Case 1b (right) where $\{a, b, e, d\}$ are vertices and $\{c, f\}$ are not.

Lemma 12 *Given the positions of $\{a, c, d, f\}$ and the preprocessing in Lemma 2, the positions of b and e can be computed in $O(n)$ time for case* 1a *and* 1b*. Similarly, given the positions of $\{a, b, d, e\}$ and the preprocessing in Lemma 2, the positions of c and f can be computed in $O(n)$ time in cases* 2a *and* 2b*.*

Theorem 2 *Given a simple polygon P and given that* Split(S)*, if it exists, is partially overlapping with $T_S\,(\text{Split(S)})$, a solution $S = (P_1, P_2)$ to Problem 1 can be found in $O(n^3)$ if and only if P can be partitioned into two congruent polygons.*

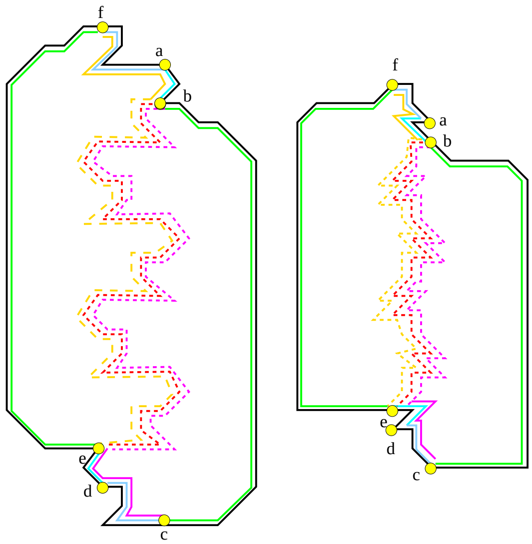

Figure 4: Case $2a$ (left) where $\{a, b, e, d\}$ are vertices and $\{c, f\}$ are not. Case $2b$ (right) where $\{a, c, d, f\}$ are vertices and $\{b, e\}$ are not.

4 Conclusion

Theorem 3 *Given a simple polygon P, we can decide if it can be partitioned into two mirror congruent polygons and find a solution $S = (P_1, P_2)$ to Problem 1, if it exists, in $O(n^3)$ time.*

References

[1] R. P. Boland and J. Urrutia. Polygon area problems. In *Canadian Conference on Computational Geometry (CCCG)*, pages 159–162, 2000.

[2] P. Eades. Symmetry finding algorithms. In G.T. Toussaint, editor, *Computational Morphology*, pages 41–51. North Holland, 1988.

[3] D. El-Khechen, T. Fevens, J. Iacono, and G. Rote. Partitioning a polygon into two congruence pieces. *Proceedings of the Kyoto International Conference on Computational Geometry and Graph Theory*, page ?, 2007.

[4] K. Erikson. Splitting a polygon into two congruent pieces. *The American Mathematical Monthly*, 103(5):393–400, 1996.

[5] M. Farach and S. Muthukrishnan. Perfect hashing for strings: formalization, algorithms and open problems. In *Symposium on Combinatorial Pattern Matching (CPM)*, 1996. `http://www.cs.rutgers.edu/~farach/pubs/tmp.html`.

[6] M. Keil. Polygon decomposition. In J.R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 11, pages 491–518. Elsevier, 2000.

[7] G. Rote. Some thoughts about decomposing a polygon into two congruent pieces. Unpublished Draft, `page.mi.fu-berlin.de/~rote/Papers/postscript/Decomposition+of+a+polytope+into+two+congruent+pieces.ps`, 1997.