Partitioning a Polygon into Two Mirror Congruent Pieces

Dania El-Khechen[*] Thomas Fevens[*] John Iacono[†]

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 (non-disjoint 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, produce an algorithm that partitions it into two proper or mirror congruent polygons P_1 and P_2, or indicate a partition is not possible with runtime polynomial in n.*

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 non-simple 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 non-simple 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.

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 $\underset{P}{\overrightarrow{ab}}$ 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 $\underset{P}{\overrightarrow{ab}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{ba}}$. Let $vd(a, b)$ be the vertical

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

[†]Department of Computer and Information Science, Polytechnic 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.

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 but is not a single line segment. In this case, 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* $\underset{P}{\overrightarrow{ab}} \overset{?}{\overset{\scriptscriptstyle MIRROR}{\cong}} \underset{P}{\overrightarrow{cd}}$ *and* $\underset{P}{\overrightarrow{ab}} \overset{?}{\overset{\scriptscriptstyle FLIP}{\cong}} \underset{P}{\overrightarrow{cd}}$ *can be answered in constant time.*

Let the length of a polyline $\underset{P}{\overrightarrow{ab}}$ (denoted $d_P(a, b)$) be the sum of the lengths of all the segments that forms this polyline.

Lemma 3 ([1]) *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. 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. Note that $d_P^{-1}(a, 0.5) = b$ is equivalent to $d_P^{-1}(b, 0.5) = a$.*

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 $\underset{P_1}{\overrightarrow{be}}$ is disjoint from the polyline $T_S\left(\underset{P_1}{\overrightarrow{be}}\right)$, $T_S\left(\underset{P_1}{\overrightarrow{be}}\right)$ partially overlaps with $\underset{P_1}{\overrightarrow{be}}$, or $\underset{P_1}{\overrightarrow{be}}$ and $\underset{P_2}{\overrightarrow{eb}}$ are line segments.*

Proof: Suppose that $T_S\left(\underset{P_1}{\overrightarrow{be}}\right) = \underset{P_2}{\overrightarrow{eb}}$. We know that by definition $\underset{P_1}{\overrightarrow{be}} \overset{\scriptscriptstyle FLIP}{\cong} \underset{P_2}{\overrightarrow{eb}}$. Therefore, the polyline $\underset{P_1}{\overrightarrow{be}}$ and its flip-congruent $\underset{P_2}{\overrightarrow{eb}}$ are mirror congruent which obviously cannot happen unless $\underset{P_1}{\overrightarrow{be}}$ and $\underset{P_2}{\overrightarrow{eb}}$ 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. A full version of the paper is appended.

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 9, 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.

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 $T_S\,(Split(S))$ and let $d = T_S(b)$, $c = T_S(e)$, $f = T_S^{-1}(b)$, and $a = T_S^{-1}(e)$.

Lemma 5 *The preprocessing described in Lemma 3 assumed, the following facts hold (see figure 1): a, b, c, d, e, f appear in clockwise order on $\delta(P)$;* $\underset{P}{\overrightarrow{fa}} \overset{\scriptscriptstyle MIRROR}{\cong} \underset{P_2}{\overrightarrow{eb}}$; $\underset{P}{\overrightarrow{cd}} \overset{\scriptscriptstyle MIRROR}{\cong} \underset{P_1}{\overrightarrow{be}}$; $\underset{P}{\overrightarrow{ab}} \overset{\scriptscriptstyle MIRROR}{\cong} \underset{P}{\overrightarrow{de}}$; $\underset{P}{\overrightarrow{bc}} \overset{\scriptscriptstyle MIRROR}{\cong} \underset{P}{\overrightarrow{ef}}$; $\underset{P}{\overrightarrow{fa}}$ *and* $\underset{P_2}{\overrightarrow{eb}}$ *are mutually flip-congurent from* $\underset{P}{\overrightarrow{cd}}$ *and* $\underset{P_1}{\overrightarrow{be}}$; $\underset{P}{\angle}\, a + \underset{P}{\angle}\, c = \underset{P_1}{\angle}\, e + \underset{P_2}{\angle}\, e$; $\underset{P}{\angle}\, f + \underset{P}{\angle}\, d = \underset{P_1}{\angle}\, b + \underset{P_2}{\angle}\, b$; *at least two of the*

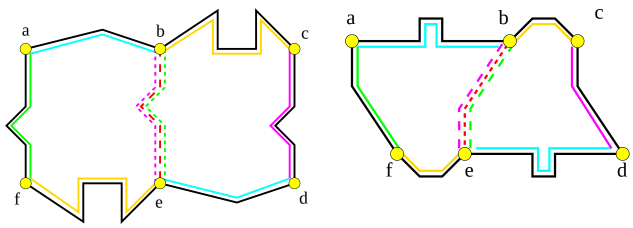

Figure 1: Polygons partitioned into two simple mirror-congruent pieces with a non-overlapping split-polyline

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 a solution* $S = (P_1, P_2)$ *of the disjoint split-polyline case of Problem 1, specified by the positions of the points* $\{a, b, c, d, e, f\}$ *as described in Lemma 5, the validity of this solution can be verified 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 9 *Given a polygon* P *and given that* $Split(S)$*, if it exists, is disjoint from* $T_S(Split(S))$*, 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 10, 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 11, we show how to verify if a given six-tuple specifies a valid solution or not. In Lemmas 12 and 13, 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 14, given that (by Lemma 10) 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.

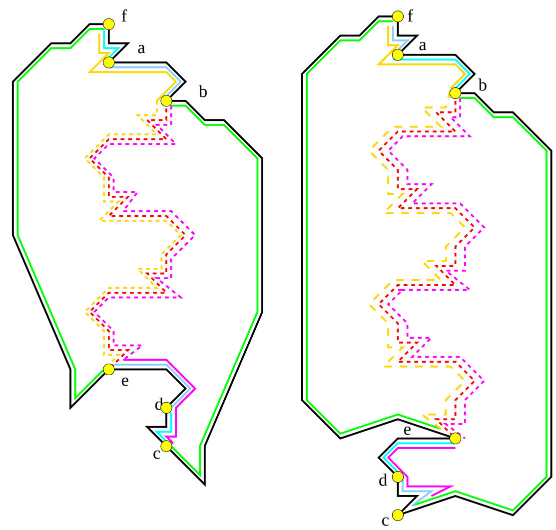

Figure 2: Polygons partitioned into two simple mirror-congruent pieces with an overlapping split-polyline

For Lemmas 10, 11, 12 and 13, assume that P can be nontrivially partitioned into two mirror congruent polygons P_1 and P_2 where $Split(S)$ is partially overlapping with $T_S(Split(S))$ and let $T_S(e) = c$, $T_S^{-1}(b) = f$.

Lemma 10 *The preprocessing described in Lemma 3 assumed, we can conclude the following facts:* $\underset{P}{\overrightarrow{ef}} \overset{MIRROR}{\cong} \underset{P}{\overrightarrow{bc}}$*;* $\underset{P_1}{\overrightarrow{fe}} \overset{MIRROR}{\cong} \underset{P_2}{\overrightarrow{cb}}$*; there exist two points* a *and* d *on* $\delta(P)$ *such that either* $\underset{P}{\overrightarrow{fa}} \overset{MIRROR}{\cong} \underset{P}{\overrightarrow{cd}}$ *and* $\underset{P}{\overrightarrow{ab}} \overset{FLIP}{\cong} \underset{P}{\overrightarrow{de}}$ *and* $\angle_P(2\pi - \angle_P d) + \angle_P f = \angle_{P_1} b + \angle_{P_2} b$ *and* $\angle_P c + \angle_P(2\pi - \angle_P a) = \angle_{P_1} e + \angle_{P_2} e$ *(this is case* 1*, see the left polygon in figure 2) or* $\underset{P}{\overrightarrow{fa}} \overset{FLIP}{\cong} \underset{P}{\overrightarrow{cd}}$ *and* $\underset{P}{\overrightarrow{ab}} \overset{MIRROR}{\cong} \underset{P}{\overrightarrow{de}}$ *and* $\angle_P d + \angle_P f = \angle_{P_1} b + \angle_{P_2} b$*,* $\angle_P c + \angle_P a = \angle_{P_1} e + \angle_{P_2} e$ *(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*; Let* $x = (vd(b, e)/vd(f, b)) \bmod vd(c, d)$*, 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* $d_P^{-1}(c, 0.5) = f$*.*

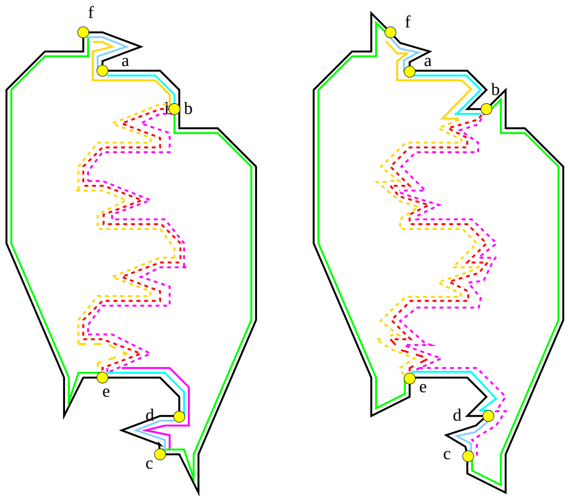

Figure 3: Left: Case 1a where $\{a, c, d, f\}$ are vertices and $\{b, e\}$ are not. Right: Case 1b where $\{a, b, e, d\}$ are vertices and $\{c, f\}$ are not

Lemma 11 *Given the preprocessing in Lemma 2 and a solution $S = (P_1, P_2)$ of the partially overlapping split-polyline case of Problem 1, specified by the positions of the points $\{a, b, c, d, e, f\}$ as described in Lemma 5, the validity of this solution can be verified in constant time.*

Lemma 12 *The points $\{a, b, c, d, e, f\}$ are as defined in Lemma 10. 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).*

Lemma 13 *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 14 *Given a polygon P and given that $Split(S)$, if it exists, is partially overlapping with $T_S(Split(S))$, 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.*

4 Conclusion

Theorem 15 *Given a polygon P, 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 mirror congruent polygons.*

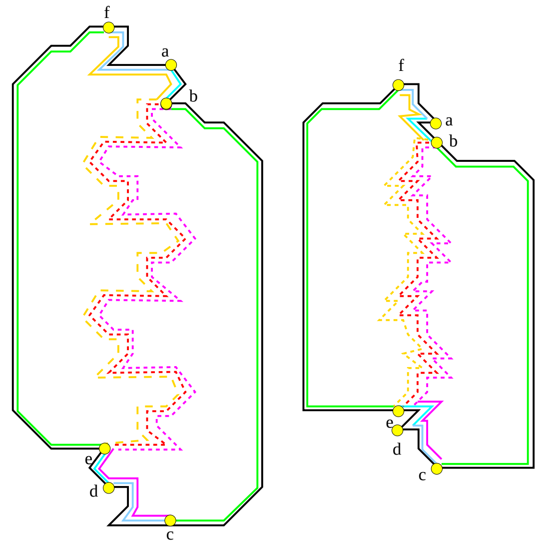

Figure 4: Left: Case 2a where $\{a, b, e, d\}$ are vertices and $\{c, f\}$ are not. Right: Case 2b where $\{a, c, d, f\}$ are vertices and $\{b, e\}$ are not

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.

Partitioning a Polygon into Two Mirror Congruent Pieces

Dania El-Khechen[*] Thomas Fevens[*] John Iacono[†]

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 (non-disjoint 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, produce an algorithm that partitions it into two (proper or mirror) congruent polygons P_1 and P_2, or indicate a partition is not possible with runtime polynomial in n.*

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 non-simple 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 non-simple 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.

2 Preliminaries

Our notions of congruence follow those in [4]. 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

[*]Department of Conputer Science and Software Engineering, Concordia University, Montréal, Québec, Canada.
[†]Department of Computer and Information Science, Polytechnic 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.

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 $\underset{P}{\overrightarrow{ab}}$ 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 $\underset{P}{\overrightarrow{ab}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{ba}}$. Let $vd(a,b)$ be the vertical distance between the two points a and b. Let $cw(a)$ and $ccw(a)$ denote respectively the segments incident to a clockwise and counterclockwise around $\delta(P)$.

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 but is not a single line segment. In this case, 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 Corollary 3.

Theorem 2 ([5]) *Given a string R of length n, an $n \times n$ table H of integers in the range $1 \ldots n^2$ can be computed in time $O(n^2)$ such that $H_{i,j} = H_{k,l}$ iff $R_{i,j} = R_{k,l}$ where $R_{i,j}$ is the substring of R from the ith to the jth character.*

Corollary 3 *Given a polygon P, with $O(n^2)$ preprocessing and space queries, of the form $\underset{P}{\overrightarrow{ab}} \overset{?}{\overset{\text{MIRROR}}{\cong}} \underset{P}{\overrightarrow{cd}}$ and $\underset{P}{\overrightarrow{ab}} \overset{?}{\overset{\text{FLIP}}{\cong}} \underset{P}{\overrightarrow{cd}}$ can be answered in constant time.*

Let the length of a polyline $\underset{P}{\overrightarrow{ab}}$ (denoted $d_P(a,b)$) be the sum of the lengths of all the segments that forms this polyline. Given a point $a \in \delta(P)$, we need to locate another $b \in \delta(P)$ such that $d_P(a,b) = x$.

Theorem 4 ([1]) *Let a* pseudo chord *denote a line segment whose endpoints are on the boundary of a polygon P. Given a simple polygon $P = v_0, \ldots, v_{n-1}$ and a query pseudo chord α, with $O(n)$ preprocessing and space, the area of the polygon P_α (determined by α such that either $v_0 \in \delta(P_\alpha)$ or $v_{n-1} \in \delta(P_\alpha)$) can be computed in constant time.*

Corollary 5 *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. 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. Note that $d_P^{-1}(a, 0.5) = b$ is equivalent to $d_P^{-1}(b, 0.5) = a$.*

3.2 Algorithms

Lemma 6 *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 $\underset{P_1}{\overrightarrow{be}}$ is disjoint from the polyline $T_S\left(\underset{P_1}{\overrightarrow{be}}\right)$, $T_S\left(\underset{P_1}{\overrightarrow{be}}\right)$ partially overlaps with $\underset{P_1}{\overrightarrow{be}}$, or $\underset{P_1}{\overrightarrow{be}}$ and $\underset{P_2}{\overrightarrow{eb}}$ are line segments.*

Proof: Suppose that $T_S\left(\underset{P_1}{\overrightarrow{be}}\right) = \underset{P_2}{\overrightarrow{eb}}$. We know that by definition $\underset{P_1}{\overrightarrow{be}} \overset{\text{FLIP}}{\cong} \underset{P_2}{\overrightarrow{eb}}$. Therefore, the polyline $\underset{P_1}{\overrightarrow{be}}$ and its flip-congruent $\underset{P_2}{\overrightarrow{eb}}$ are mirror congruent which obviously cannot happen unless $\underset{P_1}{\overrightarrow{be}}$ and $\underset{P_2}{\overrightarrow{eb}}$ 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).

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 7, 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 8, we show how to verify if a given six-tuple specifies a valid solution or not. In Lemmas 9 and 10, we show how, given two points of a solution six-tuple, we can find the rest of the points in the six-tuple. Finally, in Theorem 11, given that (by Lemma 7) 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 7, 8, 9 and 10, 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 $T_S\,(Split(S))$ and let $d = T_S(b)$, $c = T_S(e)$, $f = T_S^{-1}(b)$, and $a = T_S^{-1}(e)$.

Lemma 7 *The preprocessing described in Corollary 5 assumed, the following facts hold (see figure 1): a, b, c, d, e, f appear in clockwise order on $\delta(P)$; $\underset{P}{\overrightarrow{fa}} \overset{\text{MIRROR}}{\cong} \underset{P_2}{\overrightarrow{eb}}$; $\underset{P}{\overrightarrow{cd}} \overset{\text{MIRROR}}{\cong} \underset{P_1}{\overrightarrow{be}}$; $\underset{P}{\overrightarrow{ab}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{de}}$; $\underset{P}{\overrightarrow{bc}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{ef}}$; $\underset{P}{\overrightarrow{fa}}$ and $\underset{P_2}{\overrightarrow{eb}}$ are mutually flip-congruent from $\underset{P}{\overrightarrow{cd}}$ and $\underset{P_1}{\overrightarrow{be}}$; $\underset{P}{\angle} a + \underset{P}{\angle} c = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$; $\underset{P}{\angle} f + \underset{P}{\angle} d = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$; 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 $d_P^{-1}(c, 0.5) = f$.*

Proof: Given that $\underset{P_1}{\overrightarrow{be}}$ is a subset of $\delta(P_1)$ then its flip-congruent polyline $\underset{P_2}{\overrightarrow{eb}}$ is a subset of $\delta(P_2)$. We also know that $T_S(\underset{P_1}{\overrightarrow{be}})$ and $T_S(\underset{P_2}{\overrightarrow{eb}})$ are subsets of $\delta(P)$. Therefore, the order of $\{a, b, c, d, e, f\}$ around $\delta(P)$ is implied by T_S. Since $a = T_S^{-1}(e)$ and $f = T_S^{-1}(b)$, then the image of $\underset{P}{\overrightarrow{fa}}$ by transformation T_S is $\underset{P_2}{\overrightarrow{eb}}$. Similarly, we show that $\underset{P}{\overrightarrow{cd}} \overset{\text{MIRROR}}{\cong} \underset{P_1}{\overrightarrow{be}}$, $\underset{P}{\overrightarrow{ab}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{de}}$ and $\underset{P}{\overrightarrow{bc}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{ef}}$. Since polylines $\underset{P}{\overrightarrow{fa}}$ and $\underset{P}{\overrightarrow{cd}}$ are, respectively, mirror congruent to $\underset{P_2}{\overrightarrow{eb}}$ and $\underset{P_1}{\overrightarrow{be}}$ and since the mirror images of two flip-congruent polylines are themselves flip-congruent, then $\underset{P}{\overrightarrow{fa}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{cd}}$. Given that $a = T_S^{-1}(e)$ and $c = T_S(e)$, then $\underset{P}{\angle} a = \underset{P_2}{\angle} e$ and $\underset{P}{\angle} c = \underset{P_1}{\angle} e$. It follows that $\underset{P}{\angle} a + \underset{P}{\angle} c = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$. Similarly, we show that $\underset{P}{\angle} f + \underset{P}{\angle} d = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$. It follows that at least two of the points in $\{a, c, e\}$ and two of the points $\{b, d, f\}$ are vertices of P. Assume the preprocessing described in Corollary 5. Observe that $\underset{P}{\overrightarrow{ad}}$ is composed of $\underset{P}{\overrightarrow{ab}}$, $\underset{P}{\overrightarrow{bc}}$ and $\underset{P}{\overrightarrow{cd}}$, and $\underset{P}{\overrightarrow{da}}$ is composed of $\underset{P}{\overrightarrow{de}}$, $\underset{P}{\overrightarrow{ef}}$ and $\underset{P}{\overrightarrow{fa}}$. Hence, by the respective congruence of these polylines, given the position of a, the position of d can be found in

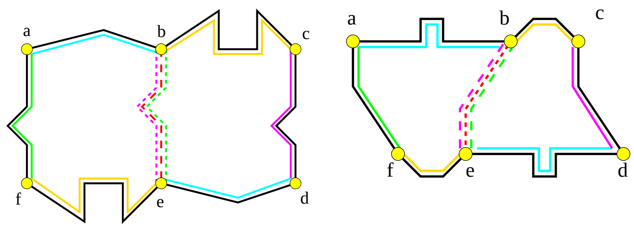

Figure 1: Polygons partitioned into two simple mirror-congruent pieces with a non-overlapping split-polyline

$O(\log n)$ time by $d_P^{-1}(a, 0.5) = d$. $\underset{P}{\overrightarrow{be}}$ is composed of $\underset{P}{\overrightarrow{bc}}$, $\underset{P}{\overrightarrow{cd}}$ and $\underset{P}{\overrightarrow{de}}$, and $\underset{P}{\overrightarrow{eb}}$ is composed of $\underset{P}{\overrightarrow{ef}}$, $\underset{P}{\overrightarrow{fa}}$ and $\underset{P}{\overrightarrow{ab}}$. Hence, by the respective congruence of these polylines, given the position of e, the position of b can be found in $O(\log n)$ time by $d_P^{-1}(e, 0.5) = b$. $\underset{P}{\overrightarrow{cf}}$ is composed of $\underset{P}{\overrightarrow{cd}}$, $\underset{P}{\overrightarrow{de}}$ and $\underset{P}{\overrightarrow{ef}}$, and $\underset{P}{\overrightarrow{fc}}$ is composed of $\underset{P}{\overrightarrow{fa}}$, $\underset{P}{\overrightarrow{ab}}$ and $\underset{P}{\overrightarrow{bc}}$. Hence, by the respective congruence of these polylines, given the position of c, the position of f can be found in $O(\log n)$ time by $d_P^{-1}(c, 0.5) = f$. $\square$

Lemma 8 *Given the preprocessing in Corollary 3 and a solution $S = (P_1, P_2)$ of the disjoint split-polyline case of Problem 1, specified by the positions of the points $\{a, b, c, d, e, f\}$ as described in Lemma 7, the validity of this solution can be verified in constant time.*

Proof: P_1 and P_2 are mirror congruent if their respective boundaries are mirror congruent. P_1 is composed (in clockwise order) of the polylines $\underset{P}{\overrightarrow{ab}}$, $\underset{P_1}{\overrightarrow{be}}$, $\underset{P}{\overrightarrow{ef}}$ and $\underset{P}{\overrightarrow{af}}$ and P_2 is composed (in clockwise order) of the polylines $\underset{P}{\overrightarrow{de}}$, $\underset{P_2}{\overrightarrow{eb}}$, $\underset{P}{\overrightarrow{bc}}$ and $\underset{P}{\overrightarrow{cd}}$. Hence, if $\underset{P}{\overrightarrow{ab}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{de}}$, $\underset{P}{\overrightarrow{bc}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{ef}}$, $\underset{P}{\overrightarrow{fa}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{cd}}$ and $|\overline{af}| = |\overline{be}| = |\overline{cd}|$, the solution is valid since the boundaries of P_1 and P_2 consist of respectively congruent polylines and copying the reversed string representation of $\underset{P}{\overrightarrow{fa}}$ onto $\overline{be}$ is possible. Otherwise, an invalid solution is reported. This verification can be done in constant time by Corollary 3. $\square$

Lemma 9 *The points $\{a, b, c, d, e, f\}$ are as defined in Lemma 7. Given the position of two points of $\{a, c, e\}$ or $\{b, d, f\}$ and the preprocessing in Corollary 5, 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.*

Proof: If (a, e) are vertices of P, then the positions of d and b are given in $O(\log n)$ time by Lemma 7. (a, e) and (b, d) form two pair of points and their respective mirror images by T_S and hence, they are sufficient to compute the glide reflection. Since $c = T_S(e)$ and $f = T_S^{-1}(b)$, c and f can then be found in constant time. Similarly, if (c, e) are vertices of P, b and f can be found in $O(\log n)$ time by Lemma 7. (b, f) and (c, e) are also two pairs of points and their mirror images and a and d can be found in constant time. If *botha* and e are not vertices and if both c and e are not vertices of P, then (a, c) are vertices and then d and f can be found in $O(\log n)$ time. Symmetrically, if pairs (b, d) or (b, f) are vertices, then we can find all six points. Else, similarly, (d, f) are vertices and we can then compute a and c in $O(\log n)$ time. However, in both cases ((a, c) or (d, f) are vertices) none of the obtained points form a pair of a point and its mirror image by T_S. Therefore, the problematic case occurs when $\{a, c, d, f\}$ are vertices of P and both b and e are not. $\square$

Lemma 10 *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 7) and the preprocessing in Corollary 3, the positions of b and e can be computed $O(n)$ time.*

Proof: Since b, in this case, is not a vertex of P, $cw(b)$ and $ccw(b)$ have the same slope. Since $T_S(ccw(f)) = cw(b)$ and $T_S^{-1}(cw(d)) = ccw(b))$, then $cw(d)$ and $ccw(f)$ have the same slope. Similarly for $cw(a)$ and $ccw(c)$. For every segment s in $\delta(P)$ such that s has the same slope as $cw(a)$ and $ccw(c)$, compute the potential b and e (the distances of b from the endpoints of the segment that contains b should be equal respectively to $|cw(a)|$ and $|ccw(c)|$ and e is half the perimeter away from b) and check, in constant time, the congruence of polylines as stated in Lemma 8. □

Theorem 11 *Given a polygon P and given that Split(S), if it exists, is disjoint from T_S (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.*

Proof: For every pair of vertices of P, verify it is (a, e) or (c, e) or (b, d) or (b, f) by computing the four remaining points as shown in Lemma 9. Verify the solution as stated in Lemma 8. If none of the previous pairs form a pair of vertices then by Lemma 9, $\{a, c, d, f\}$ are vertices and both b and e are not. For every pair of vertices of P, consider it is either (a, c) or (d, f) and compute b and e as discussed in Lemma 10. If the verification succeeds in one the cases, then copying $\underset{P}{\overrightarrow{fa}}$ onto $\overline{be}$ yields a valid partition. The "if" part is trivial. The "only if" part stems from the previous Lemmas; if the polygon is partitionable then by Lemma 7, a, b, c, d, e, f exist and appear in that order on $\delta(P)$. The split-polyline is by construction congruent to $\underset{P}{\overrightarrow{fa}}$ and $\underset{P}{\overrightarrow{cd}}$. The string matching checks for congruence of $\underset{P}{\overrightarrow{ab}}$ and $\underset{P}{\overrightarrow{de}}$ and for congruence of $\underset{P}{\overrightarrow{bc}}$ and $\underset{P}{\overrightarrow{ef}}$. Therefore, P_1 and P_2, having the same polylines defining them, are congruent. The split-polyline might however intersect $\delta(P)$. This will result in two congruent sub-polygons P_1 and P_2 that are non-simple. The algorithm clearly runs in $O(n^3)$ time. □

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 prove a sufficient condition for the periodicity of a string needed for the rest of the section. We then show the necessary conditions for the existence of a solution in Lemma 13, 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 14, we show how to verify if a given six-tuple specifies a valid solution or not. In Lemmas 15 and 16, 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. Finally, in Theorem 17, given that (by Lemma 13) 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 13, 14, 15 and 16, 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 (*Split*(S)) and let $T_S(e) = c$, $T_S^{-1}(b) = f$.

Lemma 12 *Let R be a string representing a polyline, let $m(R)$ denote the representation of the mirror image of this polyline by some transformation and let* $\mathrm{substr}(R, i, j)$ *denote a substring of R from index i to index j. Given a string R such that $R = r_1 m(r_1) r_2 r_3$ for some strings r_1, r_2 and r_3 where $|r_2| \geq |r_1|$ and such that the substrings $m(r_1) r_2 r_3$ and $r_1 m(r_1) r_2$ represents a polyline and its mirror image, then R is a periodic string with period r_1.*

Proof: Given that the substrings $m(r_1)r_2r_3$ and $r_1m(r_1)r_2$ represents a polyline and its mirror image, then:

$$m(m(r_1)r_2r_3) = r_1m(r_1)r_2$$

which implies that:

$$r_1m(r_2)m(r_3) = r_1m(r_1)r_2$$

Removing r_1, we obtain:

$$m(r_2)m(r_3) = m(r_1)r_2$$

Since $|r_2| \geq |r_1|$ and by doing the appropriate replacement of strings, we get:

$$m(r_1)\,\mathrm{substr}(m(r_2), |m(r_1)| - 1, |m(r_2)| - 1)m(r_3) = m(r_1)\,\mathrm{substr}(r_2, 0, |r_2| - |m(r_3)| - 1)m(r_3)$$

Hence,

$$\mathrm{substr}(m(r_2), |m(r_1)| - 1, |m(r_2)| - 1) = \mathrm{substr}(r_2, 0, |r_2| - |m(r_3)| - 1)$$

Therefore r_2 (and hence R) is a periodic string with period $r_1 m(r_1)$. Note that if $|R|$ is not divisible by $|r_1 m(r_1)|$, R will end with a prefix of r_1 or $m(r_1)$. □

Lemma 13 *The preprocessing described in Corollary 5 assumed, we can conclude the following facts:* $\underset{P}{\overrightarrow{ef}} \overset{\mathit{MIRROR}}{\cong} \underset{P}{\overrightarrow{bc}}$; $\underset{P_1}{\overrightarrow{fe}} \overset{\mathit{MIRROR}}{\cong} \underset{P_2}{\overrightarrow{cb}}$; *there exist two points* a *and* d *on* $\delta(P)$ *such that either* $\underset{P}{\overrightarrow{fa}} \overset{\mathit{MIRROR}}{\cong} \underset{P}{\overrightarrow{cd}}$ *and* $\underset{P}{\overrightarrow{ab}} \overset{\mathit{FLIP}}{\cong} \underset{P}{\overrightarrow{de}}$ *and* $\underset{P}{\angle}(2\pi - \underset{P}{\angle} d) + \underset{P}{\angle} f = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$ *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* $\underset{P}{\overrightarrow{fa}} \overset{\mathit{FLIP}}{\cong} \underset{P}{\overrightarrow{cd}}$ *and* $\underset{P}{\overrightarrow{ab}} \overset{\mathit{MIRROR}}{\cong} \underset{P}{\overrightarrow{de}}$ *and* $\underset{P}{\angle} d + \underset{P}{\angle} f = \underset{P_1}{\angle} b + \underset{P_2}{\angle} b$, $\underset{P}{\angle} c + \underset{P}{\angle} a = \underset{P_1}{\angle} e + \underset{P_2}{\angle} e$ *(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*; Let* $x = (vd(b, e)/vd(f, b)) \bmod vd(c, d)$*, 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* $d_P^{-1}(c, 0.5) = f$*.*

Proof: Given that $c = T_S(e)$ and $b = T_S(f)$, we conclude that $\underset{P_1}{\overrightarrow{fe}} \overset{\mathrm{MIRROR}}{\cong} \underset{P_2}{\overrightarrow{cb}}$, $\underset{P}{\overrightarrow{ef}} \overset{\mathrm{MIRROR}}{\cong} \underset{P}{\overrightarrow{bc}}$, $\underset{P}{\angle} f = \underset{P_2}{\angle} b$ and $\underset{P}{\angle} c = \underset{P_1}{\angle} e$. Since $\underset{P_1}{\overrightarrow{fe}} \overset{\mathrm{MIRROR}}{\cong} \underset{P_2}{\overrightarrow{cb}}$ and $\underset{P}{\overrightarrow{fb}}$ is a prefix of $\underset{P_1}{\overrightarrow{fe}}$, then $T_S(\underset{P}{\overrightarrow{fb}})$ is a suffix of $\underset{P_2}{\overrightarrow{cb}}$. By Lemma 12, $\underset{P_1}{\overrightarrow{fe}}$ is a periodic polyline and will be of the form $(\underset{P}{\overrightarrow{fb}} + T_S(\underset{P}{\overrightarrow{fb}}))^k + j\,\underset{P}{\overrightarrow{fb}} + r$ where $j = 0, 1$ and $k \geq 1$ (+ is the concatenation operator). Similarly, the reversed string that represents $\underset{P_2}{\overrightarrow{cb}}$ will be of the form $(T_S(\underset{P}{\overrightarrow{fb}}) + \underset{P}{\overrightarrow{fb}})^k + T_S(\underset{P}{\overrightarrow{fb}}) + r$. Note that r is a string representation of a polyline that by Lemma 12 can be any prefix of $\underset{P}{\overrightarrow{fb}}$ or $T_S(\underset{P}{\overrightarrow{fb}})$). If $j = 0$ then r is a prefix of $\underset{P}{\overrightarrow{fb}}$ (see figure 2), else if $j = 1$ then r is a prefix of $T_S(\underset{P}{\overrightarrow{fb}})$ (see figure 2). Let d' be the start point of r on $\underset{P_1}{\overrightarrow{fe}}$ and let a be the endpoint of the copy of r on $\underset{P}{\overrightarrow{fb}}$ then there exists a point d on $\underset{P_2}{\overrightarrow{cb}}$ such that $d = T_S(d')$. If $j = 0$ then $\underset{P}{\overrightarrow{cd}} \overset{\mathrm{MIRROR}}{\cong} \underset{P_1}{\overrightarrow{d'e}}$ which implies that $\underset{P}{\overrightarrow{cd}} \overset{\mathrm{MIRROR}}{\cong} \underset{P}{\overrightarrow{fa}}$, $\underset{P}{\overrightarrow{de}} \overset{\mathrm{FLIP}}{\cong} \underset{P}{\overrightarrow{ab}}$, $\underset{P_1}{\angle} b = \underset{P}{\angle}(2\pi - \underset{P}{\angle} d)$ and $\underset{P_2}{\angle} e = \underset{P}{\angle}(2\pi - \underset{P}{\angle} a)$. If $j = 1$ then $\underset{P}{\overrightarrow{cd}} \overset{\mathrm{FLIP}}{\cong} \underset{P_1}{\overrightarrow{d'e}}$ which implies that $\underset{P}{\overrightarrow{cd}} \overset{\mathrm{FLIP}}{\cong} \underset{P}{\overrightarrow{fa}}$, $\underset{P}{\overrightarrow{de}} \overset{\mathrm{FLIP}}{\cong} \underset{P}{\overrightarrow{ab}}$, $\underset{P_1}{\angle} b = \underset{P}{\angle}(2\pi - \underset{P}{\angle} d)$ and $\underset{P_2}{\angle} e = \underset{P}{\angle} a$. It follows that at least two of the points in $\{a, c, e\}$ and two of the points $\{b, d, f\}$ are vertices of P. Due to the periodicity of $\underset{P}{\overrightarrow{be}}$, we conclude that x is an odd number in case 1 and x is an even number in case 2. The order of $\{a, b, c, d, e, f\}$ around $\delta(P)$ is implied. Since $\underset{P}{\overrightarrow{ad}}$ is composed of $\underset{P}{\overrightarrow{ab}}$, $\underset{P}{\overrightarrow{bc}}$ and $\underset{P}{\overrightarrow{cd}}$, and $\underset{P}{\overrightarrow{da}}$ is composed of $\underset{P}{\overrightarrow{de}}$, $\underset{P}{\overrightarrow{ef}}$ and $\underset{P}{\overrightarrow{fa}}$, by the respective congruence of these polylines (in both cases 1 and 2), given the position of a, d can be found in $O(\log n)$ time by $d_P^{-1}(a, 0.5) = d$. Also, since $\underset{P}{\overrightarrow{be}}$ is composed of $\underset{P}{\overrightarrow{bc}}$, $\underset{P}{\overrightarrow{cd}}$ and $\underset{P}{\overrightarrow{de}}$, and $\underset{P}{\overrightarrow{eb}}$ is composed of $\underset{P}{\overrightarrow{ef}}$, $\underset{P}{\overrightarrow{fa}}$ and $\underset{P}{\overrightarrow{ab}}$, by the respective congruence of these polylines (in both cases 1 and 2), given the position of b, e can be found in $O(\log n)$ time by $d_P^{-1}(b, 0.5) = e$. Finally, since $\underset{P}{\overrightarrow{cf}}$ is composed of $\underset{P}{\overrightarrow{cd}}$, $\underset{P}{\overrightarrow{de}}$ and $\underset{P}{\overrightarrow{ef}}$, and $\underset{P}{\overrightarrow{fc}}$ is composed of $\underset{P}{\overrightarrow{fa}}$,

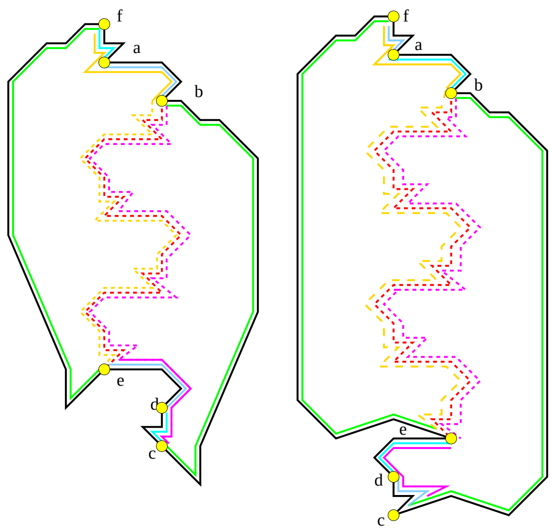

Figure 2: Polygons partitioned into two simple mirror-congruent pieces with an overlapping split-polyline

$\underset{P}{\overrightarrow{ab}}$ and $\underset{P}{\overrightarrow{bc}}$, by the respective congruence of these polylines (in both cases 1 and 2), given the position of c, f can be found in $O(\log n)$ time by $d_P^{-1}(c, 0.5) = f$. □

Lemma 14 *Given the preprocessing in Corollary 3 and a solution $S = (P_1, P_2)$ of the partially overlapping split-polyline case of Problem 1, specified by the positions of the points $\{a, b, c, d, e, f\}$ as described in Lemma 7, the validity of this solution can be verified in constant time.*

Proof: P_1 and P_2 are mirror congruent if their respective boundaries are mirror congruent. P_1 is composed (in clockwise order) of the polylines $\underset{P}{\overrightarrow{ef}}$, $\underset{P}{\overrightarrow{fa}}$, $\underset{P}{\overrightarrow{ab}}$ and $\underset{P_1}{\overrightarrow{be}}$ and P_2 is composed (in clockwise order) of the polylines $\underset{P}{\overrightarrow{bc}}$, $\underset{P}{\overrightarrow{cd}}$, $\underset{P}{\overrightarrow{de}}$ and $\underset{P_2}{\overrightarrow{eb}}$. Let $m = (vd(b, e)/vd(f, b))$. We need to consider two cases. First, if $\underset{P}{\overrightarrow{ab}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{de}}$, $\underset{P}{\overrightarrow{bc}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{ef}}$, $\underset{P}{\overrightarrow{fa}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{cd}}$ and $vd(f, b) | (vd(b, e) \bmod vd(s)$ where $s \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{cd}}$, we are in case 1. Else, if $\underset{P}{\overrightarrow{fa}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{cd}}$ and $\underset{P}{\overrightarrow{ab}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{de}}$, $\underset{P}{\overrightarrow{bc}} \overset{\text{MIRROR}}{\cong} \underset{P}{\overrightarrow{ef}}$ and $vd(f, b) | (vd(b, e) \bmod vd(s)$ where $s \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{cd}}$, we are in case 2. In both cases, the solution is valid since copying an alternation of $T(\underset{P}{\overrightarrow{fb}})$ and $\underset{P}{\overrightarrow{fb}}$, m times followed by a copy of $\underset{P}{\overrightarrow{fa}}$ (case 1) or $\underset{P}{\overrightarrow{cd}}$ (case 2) onto $\overline{be}$ will imply that $\underset{P_1}{\overrightarrow{fe}} \overset{\text{MIRROR}}{\cong} \underset{P_2}{\overrightarrow{cb}}$ and hence that the boundaries of P_1 and P_2 consist of respectively congruent polylines. Otherwise, an invalid solution is reported. This verification can be done in constant time by Corollary 3. □

Lemma 15 *The points $\{a, b, c, d, e, f\}$ are as defined in Lemma 13. Given the position of any two of $\{a, c, e\}$ or $\{b, d, f\}$ and the preprocessing in Corollary 5, 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).*

Proof: By Lemma 13, at least two of $\{a, c, e\}$ and at least two of $\{b, d, f\}$ are vertices of P. If (c, e) are vertices of P, then the position of the four remaining points can be found in the following way. f and b are

	a	b	c	d	e	f
i	-	-	V	-	V	-
ii	-	V	-	-	-	V
iii	-	Not V	Not V	-	V	V
iv	-	V	V	-	Not V	Not V
v	V	V	Not V	V	V	Not V
vi	V	Not V	V	V	Not V	V

Table 1: Sub-Cases: V stands for "is a vertex", Not V for "is not a vertex" and - for "either"

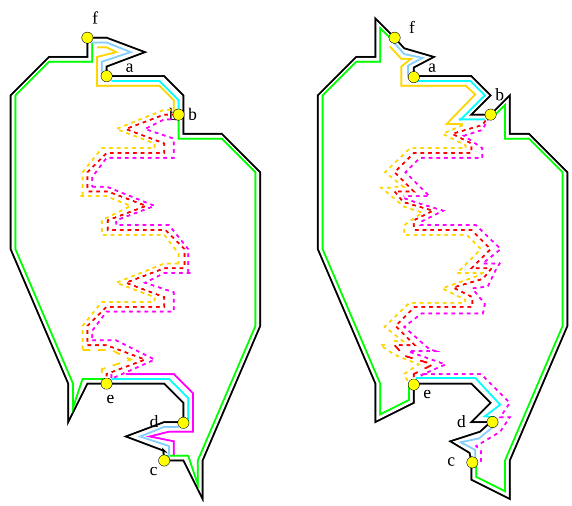

Figure 3: Left: Case 1a where $\{a,c,d,f\}$ are vertices and $\{b,e\}$ are not. Right: Case 1b where $\{a,b,e,d\}$ are vertices and $\{c,f\}$ are not

given in $O(\log n)$ time by Lemma 13. The pairs (b,f) and (c,e) form two pairs of points and their respective mirror images by T_S and hence, they are sufficient to compute the glide reflection. It remains to compute the positions of a and d. Let $d' = T_S^{-1}(d)$. By definition, d' is on the polyline $\underset{P_1}{\overrightarrow{be}}$. In case 1, see left of figure 2, d can be directly computed by translating b in the direction of the glide and the norm of the translation vector is given by $(\lfloor \frac{vd(b,e)}{vd(f,b)} \rfloor + 1)vd(f,b)$. In case 2, see right of figure 2, d' is the translate of b in the direction of the glide and the norm of translation vector is given by: $\lfloor \frac{vd(b,e)}{vd(f,b)} \rfloor vd(f,b)$. We can then compute d since its the image of d' by the glide reflection. In both cases, we can find a by $d_P^{-1}(f, d(c,d)) = a$. If (b,f) are vertices of P, finding the position of the four remaining points is similar. However, if neither the pair (c,e) nor the pair (b,f) gives us the four points, then it is easy to see by Lemma 13 and a combinatorial counting that four sub-cases remains to be considered, see table 1. Sub-cases iii and iv are similar to the sub-cases above (since if (e,f) ((b,c)) are vertices, we can compute b and c in $O(\log n)$ by Lemma 13 (e and f). Sub-cases v (where both c and f are not vertices of P) and vi (where both b and e are not vertices of P) are more complicated and are shown in both figure 3 and figure 4 for case 1 and case 2 respectively. □

Lemma 16 *Given the positions of $\{a,c,d,f\}$ and the preprocessing in Corollary 3, 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 Corollary 3, the positions of c and f can be computed in $O(n)$ time in cases 2a and 2b.*

Proof: In cases 1a and 1b, (see figure 3), given the positions of $\{a,c,d,f\}$ and the fact that $\underset{P}{\overrightarrow{ab}} \overset{\text{FLIP}}{\cong} \underset{P}{\overrightarrow{de}}$ by

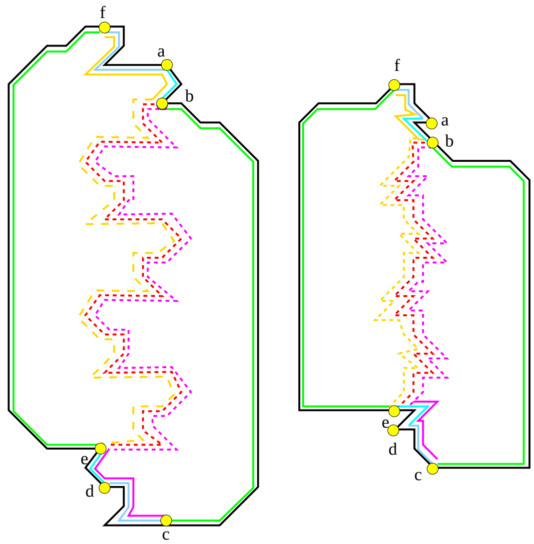

Figure 4: Left: Case $2a$ where $\{a, b, e, d\}$ are vertices and $\{c, f\}$ are not. Right: Case $2b$ where $\{a, c, d, f\}$ are vertices and $\{b, e\}$ are not

Lemma 13, we observe that $ccw(b)$ and $cw(d)$ have the same length and slope. In case $1a$, since b is not vertex then $cw(b)$ and $ccw(b)$ have the same slope. For every segment $s \in \delta(P)$ (clockwise from a) that has the same slope as $cw(d)$, compute the potential b (distance $|cw(d)|$ from the first encountered endpoint). The potential e can be computed in $O(\log n)$ by Lemma 13. In case $1b$, b is a vertex of $\delta(P)$. For every vertex $p \in \delta(P)$ (clockwise from a) such that $ccw(p)$ has the same slope and length as $cw(d)$, consider as the potential b, compute e in $O(\log n)$ by Lemma 13. In cases $2a$ and $2b$, (see figure 4), given the positions of $\{a, b, d, e\}$ and the fact that $\underset{P}{\vec{fa}} \overset{\text{FLIP}}{\cong} \underset{P}{\vec{cd}}$ by Lemma 13, we observe that $ccw(a)$ and $cw(c)$ have the same length and slope. In case $2a$, since c is not vertex then $cw(c)$ and $ccw(c)$ have the same slope. For every segment $s \in \delta(P)$ (counterclockwise from a) that has the same slope as $ccw(a)$, compute the potential c (distance $|ccw(a)|$ from the first encountered endpoint). The potential f can be computed in $O(\log n)$ by Lemma 13. In cases $2b$, c is a vertex of $\delta(P)$. For every vertex $p \in \delta(P)$ (counterclockwise from a) such that $cw(p)$ has the same slope and length as $ccw(b)$, consider as the potential c, compute f using the function d_P. In all four cases, the validity of the solution can be checked in constant time as stated in Lemma 14. □

Theorem 17 *Given a polygon P and given that $Split(S)$, if it exists, is partially overlapping with T_S $(Split(S))$, 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.*

Proof: For every pair of vertices of P, verify it is (c, e) or (b, f) or (e, f) or (b, c) by computing the remaining four points as shown in Lemma 15. For every position of a computed six-tuple, verify the validity of the solution considering both cases 1 and case 2 in constant time as stated in Lemma 14. If none of the previous pairs form a pair of vertices, then either $\{a, c, d, f\}$ (case 1) or $\{a, b, d, e\}$ (case 2) are vertices by Lemma 15. In case $1a$: for every pair of vertices, assume it is (a, c), then d and f can be computed in $O(\log n)$. Find b and e and do the verification as discussed in Lemma 14. In case $1b$: for every pair (p, s) where p is a vertex of $\delta(P)$ and s is a segment in $\delta(P)$, verify if p is a and s is the segment such that $c \in s$ (c is positioned $|ccw(a)|$ from an endpoint of s) by computing d and f in $O(\log n)$, by computing b and e and doing the verification as discussed in Lemma 14. In case $2a$: for every pair of vertices, assume it is (a, e), then d and b can be computed in $O(\log n)$. Find c and f and do the verification as discussed in Lemma 14. In case $2b$: for every pair (p, s) where p is a vertex of $\delta(P)$ and s is a segment in $\delta(P)$, verify if p is a and s is the segment such that $e \in s$ (e is positioned $|ccw(a)|$ from an endpoint of s) by computing b and d in $O(\log n)$, by computing c and f and doing the verification as discussed in Lemma 14. Let $m = vd(b,e)/vd(f,b)$. If the verification succeeds in any of the previous cases, then copying an alternation of $T_S(\underset{P}{\overrightarrow{fb}})$ and $\underset{P}{\overrightarrow{fb}}$, m times and then appending a copy of $\underset{P}{\overrightarrow{fa}}$ (case 1) or its mirror image (case 2) onto $\overline{be}$ yields a valid partition. The "if" part is trivial. The "only if" part stems from the previous Lemmas; if P is partitionable then by Lemma 13, $\{a, b, c, d, e, f\}$ exist and appear in that order on $\delta(P)$. The split-polyline allows by construction for $\underset{P_1}{\overrightarrow{fe}}$ to be congruent to $\underset{P_2}{\overrightarrow{cb}}$. The string matching checks for congruence of $\underset{P}{\overrightarrow{ef}}$ and $\underset{P}{\overrightarrow{bc}}$. Therefore, P_1 and P_2 having the same polylines defining them, are congruent. The split-polyline might however intersect $\delta(P)$. This will result in two congruent sub-polygons P_1 and P_2 that are non-simple. $\square$

4 Conclusion

Theorem 18 *Given a polygon P, 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 mirror congruent polygons.*

Proof: By Lemma 6, the split-polyline *Split*(S) is either disjoint or partially overlapping with its mirror image T_S (*Split*(S)). Hence, given a polygon P, we run the algorithm for the disjoint case from Theorem 11. If it fails, we run the algorithm for the partially overlapping case from Theorem 17. If any of the two cases succeed report the partition else report that P is not partitionable into two mirror congruent pieces. $\square$

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.