On Compatible Star Decompositions of Simple Polygons

Michal Etzion and Ari Rappoport

Abstract—We introduce the notion of *compatible star decompositions* of simple polygons. In general, given two polygons with a correspondence between their vertices, two polygonal decompositions of the two polygons are said to be *compatible* if there exists a one-to-one mapping between them such that the corresponding pieces are defined by corresponding vertices. For compatible *star* decompositions, we also require correspondence between star points of the star pieces. Compatible star decompositions have applications in computer animation and shape representation and analysis.

We present two algorithms for constructing compatible star decompositions of two simple polygons. The first algorithm is optimal in the number of pieces in the decomposition, providing that such a decomposition exists without adding Steiner vertices. The second algorithm constructs compatible star decompositions with Steiner vertices, which are not minimal in the number of pieces but are asymptotically worst case optimal in this number and in the number of added Steiner vertices. We prove that some pairs of polygons require $\Omega(n^2)$ pieces, and that the decompositions computed by the second algorithm possess no more than $O(n^2)$ pieces.

In addition to the contributions regarding compatible star decompositions, the paper also corrects an error in the only previously published polynomial algorithm for constructing a minimal star decomposition of a simple polygon, an error which might lead to a nonminimal decomposition.

Index Terms—Star decomposition, minimal star decomposition, compatible decompositions, compatible star decompositions.

1 Introduction

Decomposing a shape into simpler components is a major topic in computational geometry, having a large number of important applications. One interesting type of decompositions are *star decompositions* of polygons. A *star polygon* is a polygon for which there exists a point (a *star point*) from which all other points are visible. A *star decomposition* is a decomposition into star polygons.

There are two types of polygon decompositions: decompositions with Steiner points, and decompositions without Steiner points. A *Steiner point* is a point in the polygon (including its boundary) which is not a vertex. In a decomposition without Steiner points, all vertices of the components are exactly the polygon's vertices. In a decomposition with Steiner points, the pieces' vertices can also be interior or boundary points of the polygon. Note that every polygon can be decomposed into star pieces without adding Steiner points, since a triangulation is a valid star decomposition. However, adding Steiner points may allow a decomposition into a smaller number of pieces. There is a single published polynomial algorithm for decomposing a polygon into a minimal number of star pieces without adding Steiner points [6]. The question whether a minimal star decomposition of a polygon using Steiner points can be found in polynomial time is an open problem.

A relatively new problem category in this area of polygon decompositions is the computation of *compatible decompositions*, which belongs to the more general category of combinatorial structures involving several geometric objects instead of a single object. A *complete correspondence* between two polygons is a one-to-one mapping between their vertices that preserves vertex adjacencies and orientation. Given two polygons with a complete correspondence between their vertices, decompositions of the two polygons are *compatible* if there exists a one-to-one mapping between them such that corresponding pieces are defined by corresponding vertices. Fig. 1 shows compatible star decompositions of two polygons. Corresponding vertices are labeled with the same numbers.

Note that while one polygon can always be decomposed into triangles, convex pieces or star pieces without adding Steiner points, a pair of polygons might not possess compatible decompositions into these types of pieces without adding Steiner points. This case is illustrated in Fig. 2. The two polygons cannot be compatibly star decomposed without adding Steiner points. The polygon on the right is not a star polygon, and no diagonal of this polygon is a valid diagonal in the polygon on the left. In order to decompose the two polygons compatibly, Steiner points must be added, as shown in Fig. 2. In this case, two Steiner points were added, yielding a star decomposition having four star pieces: {0, 1, *a*, 7}, {*a*, 1, 2}, {2, 3, 4, *b*, *a*}, {*a*, *b*, 4, 5, 6, 7}.

Saalfeld [10] used compatible triangulations in order to compute a homeomorphism between the interiors of rectangular regions in the plane. Aronov et al. [1] studied compatible triangulations systematically showing how to construct compatible triangulations of arbitrary two polygons using $O(n^2)$ Steiner points. They also showed that n^2 is a lower bound: There are pairs of polygons which need $\Omega(n^2)$ Steiner points in order to be compatibly triangulated.

- *The authors are with the Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem 91904, Israel.*
WWW: http://www.cs.huji.ac.il/~arir, http://www.cs.huji.ac.il/~michals
E-mail: {michals, arir}@cs.huji.ac.il.

For information on obtaining reprints of this article, please send e-mail to: transvcg@computer.org, and reference IEEECS Log Number V96034.

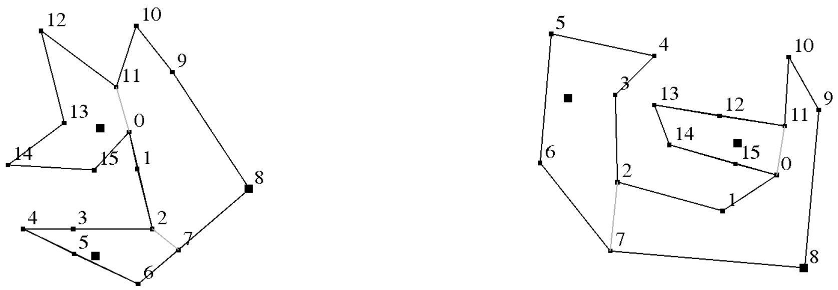

Fig. 1. Compatible star decompositions of two polygons. The black squares mark corresponding star origins.

Souvaine and Wenger [15] construct compatible triangulations of any two point sets lying in the interior of rectangles in the plane. They also use $O(n^2)$ Steiner points, proving that this result is worst case optimal. Triangles are stars, so constructing compatible triangulations answers the problem of constructing compatible star decompositions. However, letting the pieces of the decompositions be arbitrary star pieces, not just triangles, may allow smaller decompositions.

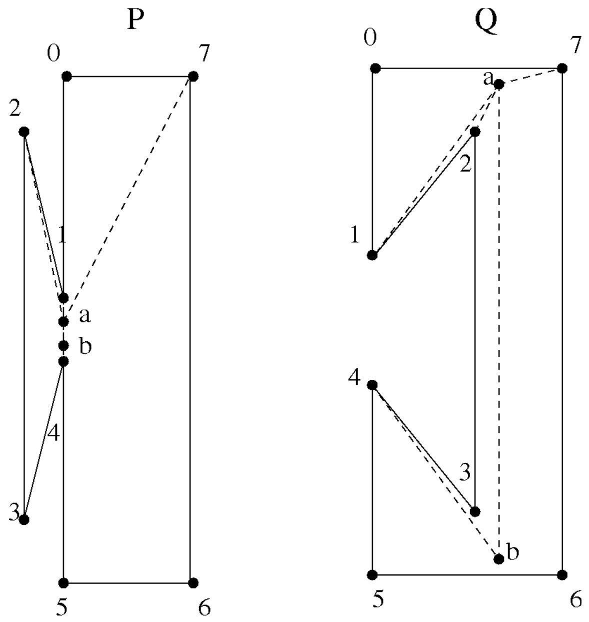

Fig. 2. Compatible star decompositions with Steiner points.

In this paper, we discuss star decompositions, and for the first time the issue of compatible star decompositions. In Section 2, we describe the algorithm in [6] for decomposing a simple polygon into minimal number of star pieces. This algorithm forms the basis of our algorithms, and we correct an error in that algorithm which might lead to non-minimal decompositions. In Section 3, we define the notion of compatible star decompositions. Section 4 presents an algorithm for constructing *minimal* compatible star decompositions if compatible decompositions exist without adding Steiner points. Section 5 presents an algorithm for constructing compatible star decompositions when it is essential to add Steiner points. Finally, in Section 6, we prove that some pairs of polygons might require decompositions with $\Omega(n^2)$ pieces, which shows that the algorithm of Section 5 is asymptotically worst-case optimal (in the size of the decomposition) even though the decompositions it creates are not minimal.

1.1 An Example Application

In [13], we presented a polygon representation, the *star-skeleton* representation, which is based on compatible star decompositions. This representation was used to blend between two polygons. Shape blending is a central problem in computer animation [11], [12], [2], [8]. Previous approaches to this problem, including direct vertex interpolation and an interpolation based on edge lengths and angles between edges [12], tend to produce self intersections and shape distortions.

Shape blending using the star-skeleton representaion is done by star decomposing the two polygons compatibly, and building a skeleton connecting the star pieces. Blending is done by interpolating the skeletons, and then unfolding the star pieces from the skeleton. Fig. 3 shows a blend sequence generated by this method. The upper part of the figure shows the compatible star decompositions of the two polygons on the right and on the left, while the lower part shows their skeletons. The algorithms for constructing compatible star decompositions presented in this paper are implemented as part of the interactive shape blending system.

Shape blending using the star-skeleton representation generates improved blend sequences, since this representation considers the interiors of the polygons, not only the boundaries, and explicitly models an interdependence between all the vertices of the polygons. The concept of compatible star decompositions is appropriate for this application since two star polygons can be fairly blended without any self intersections, minimizing shape distortions.

2 Minimal Star Decomposition of a Simple Polygon

Keil [6] presented a sophisticated algorithm for constructing a minimal star decomposition of a simple polygon without adding Steiner points. The algorithm runs in time $O(n^7)$ where n is the number of the polygon's vertices. In this section we describe this algorithm, which forms the basis of our algorithms as well. There is an error in this algorithm which might lead to a non-minimal decomposition; we give a correction here. The description of the algorithm is rather brief. The interested reader can find the details in [6] or [9]. The version of the algorithm given in [5] is correct.

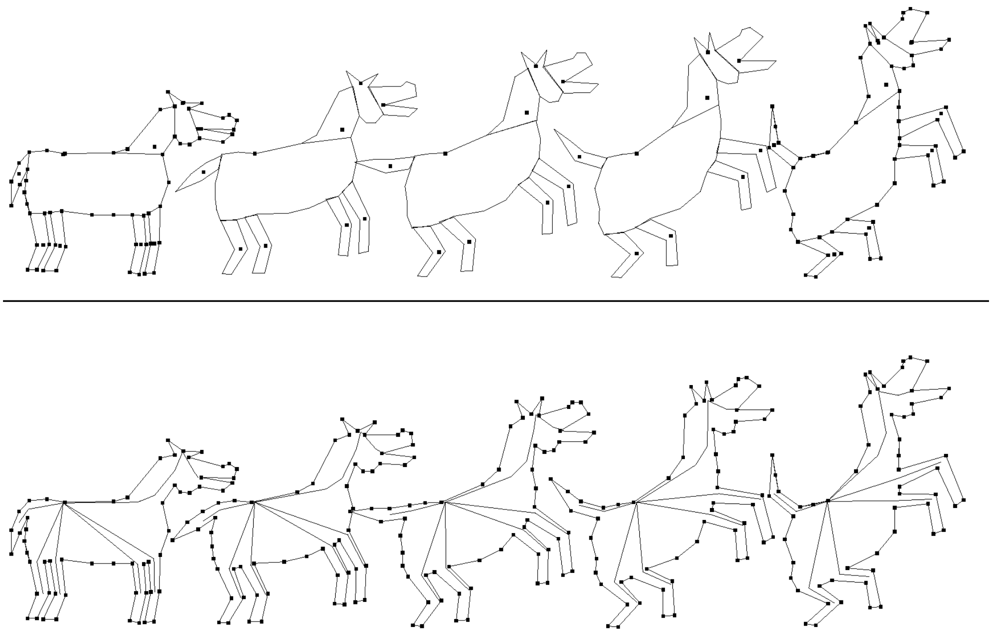

Fig. 3. Shape blending using compatible star decompositions.

The algorithm finds a minimal decomposition of a polygon among all star decompositions of the polygon. What we actually describe here is a version of the algorithm which finds the minimal decomposition among all the star decompositions *in which each star piece has a vertex which is a star point*. This version is presented because it is simpler and because our algorithm for constructing compatible star decompositions of two polygons is based on this version (Section 4). The presented version is also more efficient; it takes $O(n^4)$ time. The correction we give here is the same for both versions. From now on, the term "star decomposition" will refer to star decompositions in which each star piece has a vertex which is a star point, since we are interested only in decompositions which fulfill this additional requirement.

The algorithm is a dynamic programming algorithm. In dynamic programming an optimal solution is found by joining optimal solutions to subproblems. An optimal solution to a subproblem is represented by a state with an associated cost. The algorithm progresses in stages which solve all subproblems of a certain size using solutions computed at previous stages.

In our case, the algorithm builds a minimal decomposition of polygon P from minimal decompositions of subpolygons of P. The subpolygons P_{ij} are defined by consecutive vertices $i, \ldots, j$ where $j > i + 1$, and they exist when i sees j. In each stage s the algorithm builds the decompositions of subpolygons P_{ij} in which $j - i = s$. Each state is a decomposition of a subpolygon. The state's cost is the size of the decomposition. In each decomposition M of a subpolygon P_{ij} there is one star one of whose edges is $[j, i]$. We call this star the *base star* of the decomposition M, and label it as $S_{ij}(M)$. The triangle defined by the vertices i, m, j is denoted by T_{imj}. T_{imj} is defined only if $i < m < j$ and m sees both i and j.

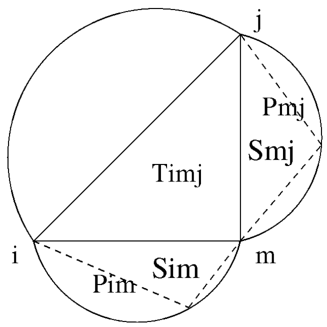

Fig. 4. A star decomposition of P_{ij} is built from star decompositions of P_{im} and P_{mj} by merging the triangle T_{imj} with the base stars S_{im} and S_{mj}.

A state is built from states from earlier stages. The decomposition M of P_{ij} is built by joining a decomposition A of P_{im} and a decomposition B of P_{mj} for some m ($i < m < j$) which sees both i and j (Fig. 4). This joining can be done in a number of ways :[1]

- *DoubleMerge*—merging T_{imj} with both base stars:
  $M(P_{ij}) = (A(P_{im}) \setminus S_{im}(A)) \cup (B(P_{mj}) \setminus S_{mj}(B)) \cup \{S_{im}(A) + S_{mj}(B) + T_{imj}\}$
  $\text{size}(M(P_{ij})) = \text{size}(A(P_{im})) + \text{size}(B(P_{mj})) - 1$
- *SingleIMerge*—merging T_{imj} only with $S_{im}(A)$:
  $M(P_{ij}) = (A(P_{im}) \setminus S_{im}(A)) \cup B(P_{mj}) \cup \{S_{im}(A) + T_{imj}\}$
  $\text{size}(M(P_{ij})) = \text{size}(A(P_{im})) + \text{size}(B(P_{mj}))$
- *SingleJMerge*—merging T_{imj} only with $S_{mj}(B)$:
  $M(P_{ij}) = A(P_{im}) \cup (B(P_{mj}) \setminus S_{mj}(B)) \cup \{S_{mj}(\mathrm{B}) + T_{imj}\}$
  $\text{size}(M(P_{ij})) = \text{size}(A(P_{im})) + \text{size}(B(P_{mj}))$

1. In the following description set union is denoted by ∪, set difference by \, and the operation of forming a single polygon by taking the point set union of other polygons is denoted by the + symbol.

- *NoMerge*—not merging T_{imj} with any of the base stars: $M(P_{ij}) = A(P_{im}) \cup B(P_{mj}) \cup \{T_{imj}\}$
size($M(P_{ij})$) = size($A(P_{im})$) + size($B(P_{mj})$) + 1

For each subpolygon P_{ij} there exist two types of states:

- $M(P_{ij})$—A minimal star decomposition of P_{ij}. There is one such state for each subpolygon. This is a minimal decomposition among all star decompositions of P_{ij}.
- $M_x(P_{ij})$—A minimal decomposition of P_{ij} *by x*. There is one such state for each combination of a subpolygon and a vertex.[2] This is a minimal decomposition of P_{ij} among all decompositions whose base star is seen from x. The base star might not be a star, i.e., x might be outside the base star. The other elements in the decomposition except the base star are all stars. Note that as a result, $M_x(P_{ij})$ is not a valid star decomposition of P_{ij} since its base star is not a star.

Why is the second type of states needed? Two decompositions A and B of a subpolygon P_{im}, which have different base stars, have different capabilities of merging. It could be, for instance, that $S_{im}(A) + T_{imj}$ is a star, while $S_{im}(B) + T_{imj}$ is not. In this case, $S_{im}(A)$ can be SingleIMerged with T_{imj}, while $S_{im}(B)$ cannot. Therefore, it is not sufficient to keep one minimal decomposition of each subpolygon. For each possible base star S, we have to keep a minimal decomposition among all decompositions whose base stars are S. Because we are interested only in stars having a vertex which is a star point, it is sufficient to keep for each pair (P_{ij}, x), a minimal decomposition of P_{ij} whose base star is seen from x. The recognition of the necessity of the states of the second type is an important contribution of Keil's algorithm.

A pseudocode of the algorithm is given in Fig. 8. Disregard for the time being the lines enclosed by brackets, since they are not part of the algorithm for one polygon. The main part of the algorithm is the dynamic programming part described above. In this part for every triple (P_{ij}, x, m), $M_x(P_{ij})$ is computed by merging decompositions of P_{im} and P_{mj}. In *NoMerge* and *SingleJMerge*, where S_{im} is *not* a part of S_{ij}, the minimal decomposition of P_{im} is merged, because the base star can be any base star, not necessarily a one that is seen from x. In *DoubleMerge* and *SingleIMerge* S_{im} *is* a part of S_{ij}, and therefore should be seen from x. Therefore $M_x(P_{im})$ is merged, and not $M(P_{im})$. The algorithm runs in time $O(n^4)$. The dynamic programming is performed in three nested "for" loops. The outmost one is repeated $O(n^2)$ times, and the other two—$O(n)$ times. Before the dynamic programming part there is a preprocessing part, in which the visibility graph of the polygon is computed. Computing the visibility graph takes $O(n^2)$ [3], [4]. Using the visibility graph, a test whether vertex i sees vertex j is done in constant time.

The algorithm in Fig. 8 is the corrected algorithm. In this algorithm, the decomposition $M_x(P_{ij})$ is built by selecting the minimal decomposition created by the four possible merges (*DoubleMerge*, *SingleIMerge*, *SingleJMerge*, *NoMerge*.) In the original algorithm, *DoubleMerge* was always preferred over the other merges, and SingleMerge was always preferred over *NoMerge*. This preference scheme might lead to a nonminimal decomposition. Because $M(P_{ij})$ can be smaller than $M_x(P_{ij})$, the decomposition created by SingleMerge might be smaller than the one created by *DoubleMerge*, and the decomposition created by *NoMerge* can be smaller than the one created by SingleMerge or *DoubleMerge*. For example, see Fig. 5. The original algorithm yields in this case a nonminimal decomposition as illustrated in the figure. One of the wrong decisions was taken when creating $M_0(P_{38})$ with $m = 7$. Instead of selecting *NoMerge* and creating a decomposition of two stars {{3, 7, 8}, {3, 4, 5, 6, 7}}, *SingleIMerge* was preferred, leading to a decomposition into three stars: {{3, 5, 7, 8}, {3, 4, 5}, {5, 6, 7}}.

The error can be corrected in a different way as suggested by Keil [7]. He suggested defining $M_x(P_{ij})$ to exist only if it is not larger than $M(P_{ij})$. We prefer to define $M_x(P_{ij})$ to exist whenever there is a decomposition of P_{ij} by x (i.e., there exists a decomposition of P_{ij} in which the base star is seen from x), and to check all four merge possibilities.

The proof that the algorithm given in Fig. 8 constructs a minimal star decomposition is similar to the proof given in [6]. The difference is in proving that if $M_x(P_{im})$, $M(P_{im})$, $M_x(P_{mj})$, and $M(P_{mj})$ are minimal, then $M_x(P_{ij})$ is minimal. This is true only if the correct merge is selected in creating $M_x(P_{ij})$. The original algorithm might fail in selecting the correct merge as explained above. The algorithm in Fig. 8 always selects the correct merge because it chooses the minimal from the four possible merges.

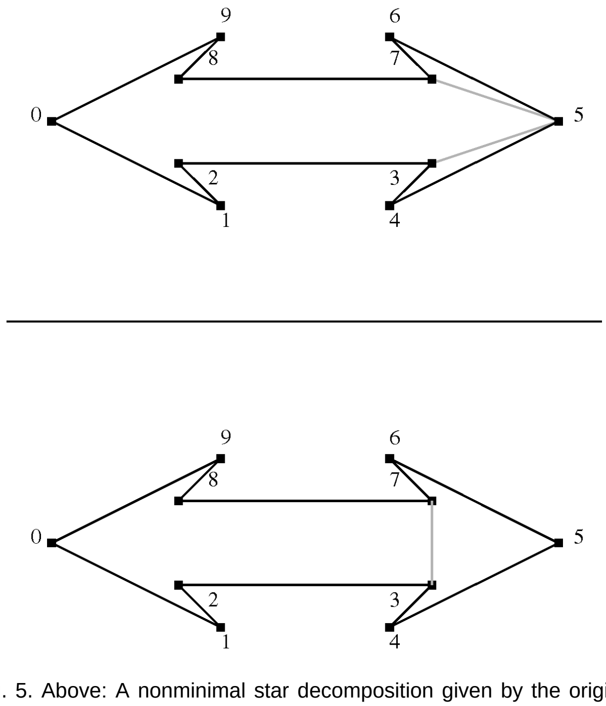

Fig. 5. Above: A nonminimal star decomposition given by the original algorithm. Below: A minimal star decomposition given by the corrected algorithm.

2. The only difference between the version described here (in which a star decomposition is legal only if each star piece has a vertex which is a star point), and the complete version (in which *all* star decompositions are legal), is the definition of x. In our case x is a vertex of the polygon, and $M_x(P_{ij})$ exists for each combination of a vertex and a subpolygon. In the complete version, x is any point from a set K of "potential" star points. In this case $M_x(P_{ij})$ exists for each combination of a point in K and a subpolygon. Keil shows how to build K of size $O(n^4)$, and proves that the kernel of any base star of a minimal decomposition of a subpolygon P_{ij} contains a point in K.

3 COMPATIBLE STAR DECOMPOSITIONS: DEFINITION

In the introduction, we stated that two decompositions are compatible if there exists a one-to-one mapping between them such that corresponding pieces are defined by corresponding vertices. This approach could have been used to define compatible star decompositions as well. However, it ignores an important feature of a star polygon, the set of its star points. Any transformation between two star pieces which utilizes their starness is based on a transformation between the star points of the two pieces. Therefore a correspondence between two star pieces should also include correspondence between the star points of the two pieces.

The star-skeleton blending method (see Section 1 and [13]) gives an example of a transformation between two star polygons, a transformation in which two star pieces are blended without self-intersections. This is achieved by choosing a pair of corresponding star points in the two pieces, and representing the vertices of each star piece in polar coordinates with respect to the chosen star points. Intermediate star polygons are generated by linearly interpolating the positions of the corresponding star points and the polar coordinates of corresponding vertices. The starness of the two pieces ensures that vertices are sorted by angle around the star point. Linear interpolation of polar coordinates of corresponding vertices preserves the order of the angles. Thus the vertices of each intermediate polygon are also sorted by angle, which means that the polygon's edges do not self-intersect.

In a transformation such as the one described above, a selected star point in one piece is transformed to the selected star point in the other piece, while the vertices of one piece are transformed to their corresponding vertices in the other piece. The star points should be chosen so that their correspondence conforms to the correspondence already defined on the two pieces, by the correspondence between their vertices. Therefore we say that two star points correspond if they are convex combinations of corresponding vertices weighted by the same weights in both polygons.

A sufficient condition for the existence of corresponding star points is the existence of a *common star vertex*. A common star vertex is a pair of corresponding vertices which are star points in the two pieces. If one such pair exists, then this pair constitutes a pair of corresponding star points. If more than one pair exist, an infinite number of corresponding star points exist: Due to the kernel's convexity, all points which are convex combinations of these vertices are star points. Note that the existence of a common star vertex is not a necessary condition for the existence of corresponding star points. However adding this condition make the algorithms for constructing compatible star decompositions simpler and more efficient.

Fig. 1 shows compatible star decompositions of two polygons. Corresponding star points are shown as black squares. In this case, the star points are the center of mass of the common star vertices. In the star defined by vertices 2-7, for instance, there are three common star vertices: 3, 5, 6, and the shown star points are the center of mass of these vertices.

The discussion above leads to the following definition of compatible star decompositions. Let P and Q be two polygons with a complete correspondence between their vertices. Let S and T be star decompositions of P and Q respectively. Assume that there exists a one-to-one mapping between the set V_S of vertices of all star pieces in S and the set V_T of vertices of all star pieces in T. If S and T do not include Steiner vertices, then this mapping is the complete correspondence between the two polygons. If S and T include Steiner vertices, then the mapping assumption means that the Steiner vertices must also possess such a mapping. In both cases we use the phrase 'a vertex in V_S corresponds to a vertex in V_T'.

Given this correspondence, we say that S and T are *compatible* if each star $s \in S$ has a mate $t \in T$ such that:

1) s and t are defined by corresponding vertices.
2) s and t have at least one common star vertex.

Given two compatible decompositions S and T of P and Q, the designation of corresponding star points in each pair of corresponding star pieces, defines a piecewise linear homeomorphism between P and Q. The homeomorphism maps every star $s \in S$ to its mate $t \in T$ in the following way: the chosen star point is mapped to its corresponding star point, and every vertex is mapped to its corresponding vertex. A *wedge* of a star is a triangle defined by the star point and two consecutive vertices of the star. Two wedges of s and t correspond if the star points and the vertices defining them correspond. The mapping of the vertices and the star points induces a unique linear map between corresponding wedges of s and t.

In the next sections we will see how to build compatible star decompositions for a pair of polygons. We assume that a complete correspondence between the two polygons is given.

4 COMPATIBLE STAR DECOMPOSITIONS WITHOUT STEINER POINTS

First, we consider compatible star decompositions of pairs of polygons which can be star-decomposed compatibly without adding Steiner points. In this section, we give an algorithm which constructs *minimal* compatible star decompositions in this case.

The algorithm is based on the algorithm of [6] for constructing a minimal star decomposition of one polygon, a special version of which was described in Section 2. The version we use constructs a minimal star decomposition in which each star piece has a vertex which is a star point. This additional requirement makes the algorithm simpler and more efficient: $O(n^4)$ instead of $O(n^7)$. Because compatible star decompositions require the existence of a common star vertex, we are in any case not interested in decompositions in which there are star pieces none of whose vertices is a star point.

Given two polygons P and Q, minimal compatible star decompositions can be found by running the algorithm described in Section 2 for P, avoiding decompositions which are not legal compatible star decompositions of P and Q.

A naive extension of the algorithm described in Section 2 can be done by substituting every visibility check in P with a visibility check in P and in Q. The extended algorithm would return only valid compatible star decompsitions. However, this algorithm excludes legal decompositions, hence it might

1) decide that no valid compatible decomposition exists when in fact one does exist, and
2) generate a nonminimal decomposition.

Fig. 6 demonstrates that when using the naive extension nonminimal decompositions can be obtained. The algorithm described below excludes only illegal decompositions, and therefore gives the minimal compatible star decompositions.

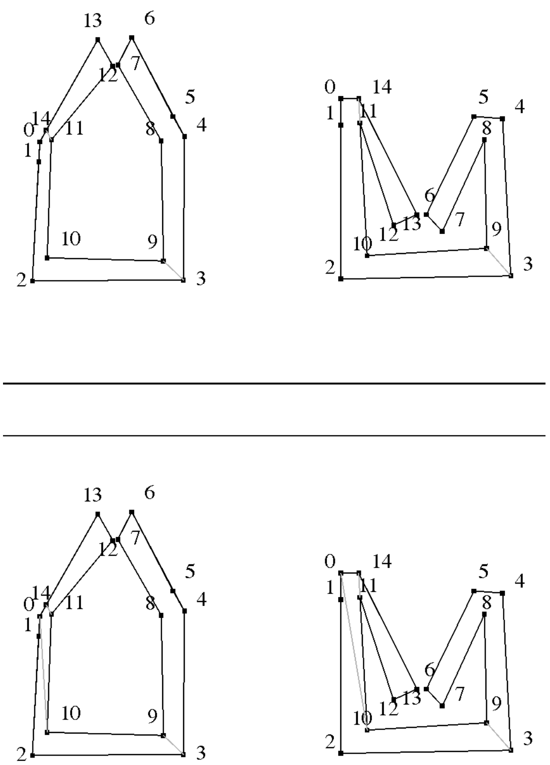

Fig. 6. Above: Minimal compatible star decompositions obtained by our algorithm. Below: Nonminimal compatible star decompositions obtained by a naive extension of the algorithm for star decomposition of one polygon.

A star decomposition of P induces a decomposition of Q defined by corresponding vertices. It cannot be viewed as compatible star decompositions of P and Q iff one of the following applies:

1) The decomposition is not legal in Q, i.e., it includes a segment which is not a diagonal in Q.
2) The requirement of a common star vertex is not fulfilled, i.e., there is a piece in the decomposition which is not wholly seen by corresponding vertices in the two polygons.

Segments are added to the decomposition only when we "close" a subpolygon, i.e. a segment $[i, j]$ is added only when we select a merge that uses $M(P_{ij})$, not $M_x(P_{ij})$. Therefore, in order to avoid decompositions of the first type, we define $M(P_{ij})$ to exist only if i sees j in Q.

In order to avoid decompositions of the second type, we do not allow base stars which are not seen by corresponding vertices. Therefore, we avoid decompositions $M_x(P_{ij})$ in which x does not see $S_{ij}(M)$ in Q. This is obtained by picking only merges which create decompositions $M_x(P_{ij})$ whose base stars are seen by x in Q. That means we pick *SingleIMerge* only if x sees j in Q, we pick *SingleJMerge* only if x sees i in Q, and we pick *NoMerge* only if x sees i, m, j in Q. One should observe that these tests are sufficient. Look for instance at *SingleIMerge*. *SingleIMerge* is picked only if $M_x(P_{im})$ exists. The existence of $M_x(P_{im})$ implies that x sees the base-star of $M_x(P_{im})$ in Q, which means that x sees in Q all the vertices of the base star of $M_x(P_{ij})$ except j.

Avoiding decompositions of the two types described above leaves only decompositions which are legal compatible star decompositions (decompositions of the type $M(P_{ij})$) and decompositions which can be merged to create legal compatible star decompositions (those of the type $M_x(P_{ij})$.) A pseudocode for the extended algorithm is found in Fig. 8. The lines enclosed by brackets are the additions for the case of two polygons. Note that P_{ij} and T_{imj} are defined as they were defined in the case of a single polygon, i.e. P_{ij} exists if i sees j in P, even if i does not see j in Q,

The algorithm in Fig. 8 finds minimal compatible star decompositions of the two input polygons P and Q if one exists. The proof that the algorithm returns minimal compatible decompositions is based on the proof in the case of a single polygon and is given in [14]. The running time of the algorithm is $O(n^4)$, like in the case of a single polygon (see Section 2).

5 Compatible Star Decompositions with Steiner Points

As we saw in Fig. 2, there are pairs of polygons which cannot be star-decomposed compatibly without adding Steiner points. We show now how to construct compatible star decompositions in these cases using no more than $O(n^2)$ Steiner points. Section 6 proves that this is asymptotically worst-case optimal in the sense that there are pairs of polygons which cannot be star decomposed compatibly into less than $\Omega(n^2)$ star pieces, which also means that they cannot be star decomposed compatibly using less than $\Omega(n^2)$ Steiner points.

Aronov et al. [1] showed how to construct compatible triangulations of two polygons using $O(n^2)$ Steiner points. Since every compatible triangulation establish a compatible star decomposition, they already showed how to construct compatible star decompositions using $O(n^2)$ Steiner points. However, the construction described below adds on average far fewer Steiner points than the construction described in [1]. This statement is both based on empirical observations and intuitive: It is reasonable to expect that when Steiner points are allowed the number of star pieces in a decomposition will be smaller than the number of triangles in a triangulation.

The construction of compatible star decompositions of two polygons P and Q is done in two steps. First P and Q

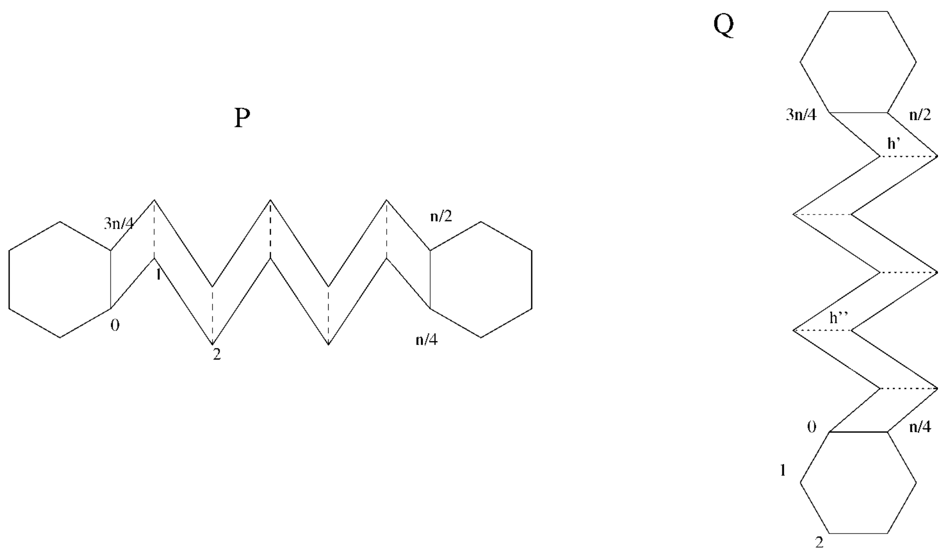

Fig. 7. Two polygons which cannot be compatibly decomposed into less than $\Omega(n^2)$ star pieces.

are decomposed into corresponding subpolygons using Steiner points, and then each pair of these subpolygons is star decomposed compatibly. In order to guarantee that each pair created in the first step possesses compatible star decompositions without adding more Steiner points, the subpolygons of P are all convex. If G is a convex subpolygon of P, and H is the corresponding subpolygon of G in Q (i.e., G and H are defined by corresponding vertices), then every legal star decomposition of H induces a legal star decomposition of G. This is because every segment connecting two vertices in G is a diagonal of G which decomposes G into two convex pieces. Moreover, because every resulting star piece is convex in G, every vertex of each star of G is a star point, and, therefore, every pair of corresponding stars in G and in H have a common star vertex. If a convex piece G cannot induce a corresponding one in Q (when its edges are not polygon edges or diagonals in Q), Steiner vertices are added to both polygons to create equivalent pieces G and H.

The algorithm itself is as follows. We compute a minimal *convex* decomposition of P [6]. At least one convex piece G in this decomposition contains only consecutive vertices $P_i, \ldots, P_j$ of P. If $[P_i, P_j]$ is not a diagonal in Q, the piece G cannot induce a legal piece in Q. In this case, a sequence of Steiner points is added. Let $s_1, ..., s_k$ ($k = j - i - 1$) be the points in Q which result from offsetting each of $Q_{i+1}, \ldots, Q_{j-1}$ by a small amount towards the inside of Q. (Points a and b in Fig. 2 result from offsetting vertices 2 and 3 towards the inside of Q.) The sequence of Steiner points added in Q is a subsequence of the above sequence which fulfills the following condition: every two consecutive points in the sequence $Q_i, s_1, \ldots, s_k, Q_j$ correspond to vertices which see each other, but *no three* consecutive points in this sequence correspond to vertices which see each other. In this manner, a Steiner point is added only if it "separates" two convex pieces in Q, thus minimizing the size of the resulting decomposition (for example, the three convex pieces in a minimal convex decomposition of Q in Fig. 2 are separated by the two Steiner points a and b.) The added Steiner vertices induce corresponding vertices in P, $s_1', \ldots, s_k'$, which are created on the diagonal $[P_i, P_j]$ in proportional distances to the distances in Q. Thus, the diagonal $[P_i, s1', ..., sk', P_j]$ in P is matched to the piecewise linear path $[Q_i, s_1, ..., s_k, Q_j]$ in Q, and H is defined. Because the Steiner points in P are added on the diagonal $[P_i, P_j]$, G refined by the added Steiner points is actually the same piece as it was before the refinement, and in particular it maintains its convexity. In Fig. 2, the convex piece of P defined by the vertices {1, 2, 3, 4} cannot be induced on Q, since the segment [1, 4] is not a diagonal in Q. The Steiner points a and b were added in both polygons, leading to two corresponding pieces defined by the vertices {1, 2, 3, 4, b, a}.

Once defined, the pieces G and H are removed from P and Q and the process continues in the same manner. In order to prove that this process yields legal decompositions of P and Q, it is enough to prove that if P and Q are simple polygons (which do not intersect themselves), then G and H are legal subpolygons of them. If this applies, then in every stage the remaining P and Q are legal simple polygons, and thus the G and H created in every stage are legal subpolygons. This implication carries inductively until all convex pieces of P are induced. It is obvious that G is a legal subpolygon of P since it is a convex piece of P which was refined by adding the Steiner vertices. H is legal because the edges of H are either edges of Q or edges between the Steiner points we added. Because every two consecutive Steiner points are epsilon-offsets of vertices which see each other, we can always find an epsilon such that the edges connecting the Steiner points will not intersect themselves or Q.

After all convex pieces of P were induced on Q, each pair of corresponding pieces is star decomposed compatibly. This is done by constructing a minimal star decomposition of H, and inducing it on G. Due to G's convexity every star decomposition of H creates legal compatible star decompositions of G and H.

Let r be the number of reflex vertices in P. In inducing each convex piece of P on Q, no more than $O(n)$ Steiner points are added. There are no more than $O(r)$ convex pieces in P. Thus, during the whole algorithm no more than $O(r \times n)$ Steiner points are added. The algorithm runs in $O(C \times S)$ time, where C is the number of convex pieces and S is the time for a star decomposition.

6 Worst-Case Optimality

In their work on compatible triangulations of simple polygons, Aronov et al. [1] proved that the two polygons in Fig. 7 cannot be compatibly triangulated using less than $\Omega(n^2)$ Steiner points. We use a similar proof to prove that these two polygons cannot be compatibly decomposed into less than $\Omega(n^2)$ star pieces.

For convenience, we will assume that n is a multiple of 4. Refer to Fig. 7. Let S and T be compatible star decompositions of P and Q, respectively. We will denote the vertical dashed lines in P by V_P and the horizontal dotted lines in Q by H_Q. The decompositions S and T induce a one-to-one piecewise-linear mapping between P and Q. By this mapping, every line in V_P is mapped to a piecewise linear path in Q. We call this group of paths V_Q. Every path in V_Q intersects $n/4$ lines in H_Q. There are $n/4$ lines in V_P, leading to $n^2/16$ intersecting (V_Q,H_Q) pairs.

We will show now that for every star piece t of the decomposition T there are no more than nine (V_Q, H_Q) pairs that intersect in t. Let h', h'' be two lines of H_Q, distanced by at least two other H_Q lines (Fig. 7). There does not exist any point in Q which sees both a point in h' and a point in h''. This means that t cannot intersect both h' and h'', which means it can intersect no more than three H_Q lines. Similarly, s, the mate of t in S, cannot intersect more than three V_P lines. Therefore, t cannot intersect more than three V_Q paths. Because t intersects at most three H_Q lines and three V_Q paths, there cannot be more than nine (H_Q, V_Q) pairs intersecting in t.

We saw that there are $n^2/16$ intersecting (H_Q,V_Q) pairs, and no more than nine pairs intersect in each star piece. This means that S, T must include $\Omega(n^2)$ star pieces.

7 Conclusion

Combinatorial structures involving several geometric objects instead of a single object form a relatively new area of activity in computational geometry. In this paper, we introduced the notion of compatible star decompositions, whose definition includes the nontrivial requirement of common star vertices. We presented two algorithms for constructing compatible star decompositions of two simple polygons. The first algorithm is optimal in the number of pieces in the decomposition, providing that such a decomposition exists without adding Steiner vertices. The second algorithm constructs compatible star decompositions with Steiner vertices, which are asymptotically worst case optimal in the number of pieces, and in the number of added Steiner vertices. We have also corrected an error in the only published algorithm for optimal decomposition of a single polygon into star pieces.

In the companion paper [13], we have shown an example for an application of compatible star decompositions to computer animation. Both algorithms presented in the present paper have been fully implemented in an interactive program which lets users draw polygons and request their decomposition. We believe that such integration of applied and theoretical research greatly contributes to both areas: applications obtain the power of sophisticated concepts and efficient algorithms, and theory obtains interesting problems and a stronger basis of belief in the correctness of algorithms.

Preprocessing:
Compute the visibility graph of P.
[Compute the visibility graph of Q.]
Dynamic Programming:
for every P_{ij} in ascending order of $j - i$
  for every vertex x compute $M_x(P_{ij})$ as follows:
    for every T_{imj}
      if $M_x(P_{im})$ exists **and** $M_x(P_{mj})$ exists
        then size(DoubleMerge)=size($M_x(P_{im})$) + size($M_x(P_{mj})$) − 1;
      if $M_x(P_{im})$ exists **and** x sees j in P
        [**and** x sees j in Q **and** $M(P_{mj})$ exists]
        then size(SingleIMerge)=size($M_x(P_{im})$) + size($M(P_{mj})$);
      if $M_x(P_{mj})$ exists **and** x sees i in P
        [**and** x sees i in Q **and** $M(P_{im})$ exists]
        then size(SingleJMerge)=size($M(P_{im})$) + size($M_x(P_{mj})$);
      if x sees i, m, j in P
        [**and** x sees i, m, j in Q]
        [**and** $M(P_{im})$ exists **and** $M(P_{mj})$ exists]
        then size(NoMerge)=size($M(P_{im})$) + size($M(P_{mj})$) + 1;
      find the minimal decomposition M of the above.
      if it is smaller than $M_x(P_{ij})$,
        then assign $M_x(P_{ij}) = M$.
  [**if** i sees j in Q]
    $M(P_{ij}) = min_x\{M_x(P_{ij}) \mid x \in S_{ij}(M_x)\}$
[**if** M_{1n} does not exist, return 'failure'.]
return (M_{1n}).

Fig. 8. The algorithm without the lines enclosed by square brackets is the corrected algorithm for constructing a minimal decomposition of one polygon P without adding Steiner vertices. With the lines enclosed by square brackets, this is the algorithm given in Section 3 for constructing minimal compatible star decompositions of P and Q without adding Steiner vertices.

Acknowledgments

We thank Mark Keil and Joe O'Rourke for their cooperation, and the reviewers for their detailed comments.

REFERENCES

[1] B. Aronov, R. Seidel, and D. Souvaine, "On Compatible Triangulations of Simple Polygons," *Computational Geometry: Theory and Applications*, vol. 3, no. 1, pp. 27-36, 1993.

[2] E.W. Bethel and S.P. Uselton, "Shape Distortion in Computer-Assisted Keyframe Animation," *Computer Animation '89*, Magnenat-Thalmann and Thalmann, eds., pp. 215-224. Tokyo: Springer, 1989.

[3] B. Joe and R.B. Simpson, "Visibility of a Simple Polygon from a Point," technical report, Univ. of Waterloo, 1985.

[4] B. Joe and R.B. Simpson, "Corrections to Lee's Visibility Polygon Algorithm," *BIT*, vol. 27, pp. 458-473, 1987.

[5] M. Keil, "Decomposing Polygons into Simpler Components," Technical Report #163/83, Dept. of Computer Science, Univ. of Toronto, 1983.

[6] M. Keil, "Decomposing a Polygon into Simple Components," *SIAM J. Computing*, vol. 14, pp. 799-817, 1985.

[7] M. Keil, Personal communication, 1993.

[8] J.R. Kent, W.E. Carlson, and R.E. Parent, "Shape Transformation for Polyhedral Objects," *Computer Graphics*, vol. 26, pp. 47-54, SGGRAPH '92.

[9] J. O'Rourke, *Art Galleries Theorems and Algorithms*. Oxford Univ. Press, 1987.

[10] A. Saalfeld, "Joint Triangulations and Triangulation Maps," *Proc. Third ACM Symp. Computational Geometry*, pp. 195-204, Waterloo, Canada, June 1987.

[11] T.W. Sederberg and E. Greenwood, "A Physically Based Approach to 2D Shape Blending," *Computer Graphics*, vol. 26, pp. 25-34, SIGGRAPH '92.

[12] T.W. Sederberg, P. Gao, G. Wang, and H. Mu, "2D Shape Blending: An Intrinsic Solution to the Vertex Path Problem," *Computer Graphics*, vol. 27, pp. 15-18, SIGGRAPH '93.

[13] M. Shapira Etzion and A. Rappoport, "Shape Blending Using the Star-Skeleton Representation," *IEEE Computer Graphics and Applications*, vol. 15, no. 2, pp. 44-50, Mar. 1995.

[14] M. Shapira Etzion and A. Rappoport, "On Compatible Star Decompositions," technical report, Inst. of Computer Science, The Hebrew Univ. of Jerusalem, 1994.

[15] D. Souvaine and R. Wenger, "Constructing Piecewise Linear Homeomorphisms," manuscript, 1994.

Michal Etzion obtained an MSc in computer science and a BSc in mathematics and computer science from the Hebrew University of Jerusalem. She is currently a PhD student at the Institute of Computer Science of the Hebrew University of Jerusalem, Israel. Her research interests include computational geometry, geometric and solid modeling, and computer graphics.

Ari Rappoport obtained his BSc in mathematics and computer science and his PhD in computer science from the Hebrew University of Jerusalem, Israel, in 1990. He is a lecturer at the Institute of Computer Science of the Hebrew University of Jerusalem and has visited the Interactive Geometric Modeling Group at the IBM T.J. Watson Research Center in Yorktown Heights, New York. His research interests include geometric modeling, solid modeling, and computer graphics.