International Journal of Image and Graphics
Vol. 13, No. 1 (2013) 1350002 (28 pages)

DOI: 10.1142/S0219467813500022

World Scientific
www.worldscientific.com

A PROVABLE ALGORITHM TO DETECT WEAK SYMMETRY IN A POLYGON

MAHMOUD MELKEMI* and FREDERIC CORDIER†
Laboratoire de Mathématiques Informatique et Applications
Université Haute Alsace
4 Rue des Frères, Lumière 68093, Mulhouse, France
**mahmoud.melkemi@uha.fr*
†*frederic.cordier@uha.fr*

NICKOLAS S. SAPIDIS
Department of Mechanical Engineering
University of Western Macedonia
Bakola and Sialvera Street, Kozani GR-50100, Greece
nsapidis@uowm.gr

Received 26 March 2012
Revised 21 September 2012
Accepted 18 October 2012
Published 10 April 2013

This paper deals with the problem of detecting "weak symmetry" in a polygon, which is a special bijective and continuous mapping between the vertices of the given polygon. An application of this work is the automatic reconstruction of 3D polygons symmetric with respect to a plane from free-hand sketches of weakly-symmetric 2D polygons. We formalize the weak-symmetry notion and highlight its many properties which lead to an algorithm detecting it. The closest research work to the proposed approach is the detection of skewed symmetry. Skewed symmetry detection deals only with reconstruction of planar mirror-symmetric 3D polygons while our method is able to identify symmetry in projections of planar as well as nonplanar mirror-symmetric 3D polygons.

Keywords: Symmetry; curve reconstruction; computer graphics; sketching interface; computational geometry.

1. Introduction

Sketches are an efficient way to visualize an idea in conceptual design. Most people have natural facility to draw freehand sketches. In addition, drawing comprehension appears to be an inherent part of human perception. Sketching interfaces have been proven to be very helpful for users to create three-dimensional (3D) shapes quickly and easily.[1,14]

*Corresponding author.

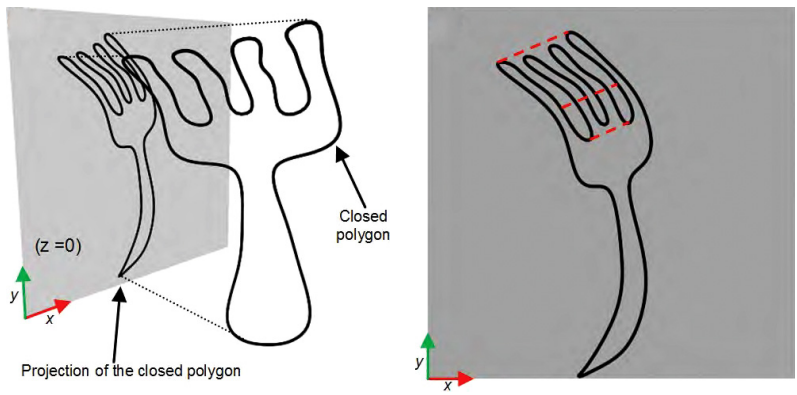
The problem of 3D reconstruction from freehand sketches is defined as follows: Given a set of two-dimensional (2D) polygons provided by the user, the 3D reconstruction consists of computing the 3D polygons (planar or nonplanar polygonal curves) such that their orthogonal projection matches the input polygons. 3D reconstruction is a difficult problem, as for each vertex of a 2D polygon, there exist an infinite number of 3D vertices whose orthogonal projection matches this 2D vertex. In this paper, we consider the reconstruction of mirror-symmetric 3D polygons. Mirror-symmetric 3D polygons are orthogonally symmetric with respect to a central plane (also known as a symmetry plane).

The problem of reconstructing mirror-symmetric 3D polygons from their orthogonal projection involves two steps: (1) finding pairs of "corresponding" vertices in the given 2D polygons, and (2) using this "correspondence" to compute vertex positions for the mirror-symmetric 3D polygon.

1.1. *Problem statement*

Let $V = \{v_0, \ldots, v_i, \ldots, v_{n-1}\}$ and $V' = \{v'_0, \ldots, v'_i, \ldots, v'_{n-1}\}$ be two sets of 3D vertices, each vertex v'_i being the mirror image of v_i. The straight lines that go through the pairs of symmetric vertices are perpendicular to the symmetry plane and are parallel to each other as well. Let $V_p = \{v_{p,0}, \ldots, v_{p,i}, \ldots, v_{p,n-1}\}$ and $V'_p = \{v'_{p,0}, \ldots, v'_{p,i}, \ldots, v'_{p,n-1}\}$ be the orthogonal projection of V and V' respectively. It follows that the lines that go through pairs of vertices $(v_{p,i}, v'_{p,i})$ are parallel to each other (see Fig. 1).

Using this property, we define the *weak-symmetry* notion. *Weak symmetry* means that there is a special correspondence among the vertices of a 2D polygon which is the orthogonal projection of a 3D mirror-symmetric polygon. Given a straight line



(a) (b)

Fig. 1. (Color online) In (a), the symmetric closed polygon and its orthogonal projection onto the plane ($z = 0$). In (b), the orthogonal projection of the symmetric polygon; lines joining pairs of symmetric vertices (red dashed lines in the figure) are parallel to each other.

ℓ, a polygon P is weakly-symmetric with respect to ℓ if only if it complies with the following properties:

(i) **Bijection.** Each point x on P, has a unique correspondent (we say symmetric) point $x' \in P$ such that the segment $[xx']$ is parallel to the line ℓ.
(ii) **Continuity.** Each point of P in a small neighborhood of a point $x \in P$ has a symmetric point in a small neighborhood of $x' \in P$, which is the symmetric of x.

Thus, P is said *weakly-symmetric* when it is weakly-symmetric with respect to at least one straight line (the formal definitions and examples are given in Sec. 3).

Given a 2D polygon, which is an orthogonal projection of an unknown mirror-symmetric 3D polygon, reconstruction of this 3D polygon is accomplished as follows:

(i) Find two sets of vertices V_p and V'_p that partition the vertices of the 2D polygon so that the vertices of V_p are weakly symmetric to those of V'_p. Our paper deals with the problem of computing these two sets. This step is the most difficult part of the reconstruction process.
(ii) Knowing V_p and V'_p the computation of the vertices of V and V' is straightforward. This step will be explained in Sec. 6.

1.2. *Related works*

1.2.1. *Reconstruction of mirror-symmetric models*

Several researchers have worked on the 3D reconstruction of mirror-symmetric models from sketches.[15] The reconstruction method proposed by Cheon *et al.*[2] uses a predefined template and assumes that the input sketch is topologically identical to the predefined template. Li *et al.*[3] have proposed a computational model that uses planarity and compactness constraints to recover 3D symmetric objects from 2D images. They assume known correspondence of symmetric points. Jiang *et al.*[4] have proposed an interactive method to create symmetric architectural structure. Their method is mostly focusing on modeling of buildings. In addition, it requires user interaction to specify the camera calibration and the geometric features of the building. Tolba *et al.*[5] describe a tool with which a user can draw a scene with 2D strokes and then he/she can visualize it from different points of view. 3D reconstruction is achieved by aligning 2D curves on a perspective grid. Cohen *et al.*[6] propose another sketching interface for 3D curve modeling, where the user can model a nonplanar curve by drawing it from a single viewpoint and its shadow on the floor plane.

1.2.2. *Symmetry detection*

In this paper, we focus on the problem of symmetry detection. To the best of our knowledge, symmetry detection in the orthogonal projection of nonplanar

mirror-symmetric 3D polygons remains an open problem. The closest research work to our approach is detection of skewed symmetry. Skewed symmetry, as defined by Kanade,[7] depicts a mirror-symmetric planar curve viewed from some (unknown) viewing direction. Posch[8] has proposed an algorithm for skewed symmetry detection. This algorithm first finds all segments parallel to the same direction and connecting pairs of symmetric vertices. The skewed symmetry is then detected by checking if the midpoints of these segments are aligned. Shen *et al.*[9,10] have proposed an algorithm based on an affine-invariant-shape representation. They first build a similarity matrix for the vertices of the curves and use this matrix to detect the lines corresponding to the skewed-symmetry axis. Yip[11] has also proposed an approach to detect skewed symmetry axes using Hough transformation.

Compared to these previous works, our approach is able to find symmetry for the projection of planar and nonplanar mirror-symmetric 3D polygons, see Fig. 1. In previous works, skewed symmetry detection is achieved by finding the symmetry axis. In the case of nonplanar mirror-symmetric 3D polygons, such an axis does not exist. Thus, these previous works cannot be used to identify symmetry.

1.3. *Overview*

To the best of our knowledge, the proposed algorithm reconstructing nonplanar mirror-symmetric 3D polygons from the 2D polygons of a freehand sketch is the first one for this problem with provable guarantee. Our algorithm produces an exact solution and ensures finding all mirror-symmetric 3D polygons that can be reconstructed from a given 2D polygon. We highlight and formalize a new notion of symmetry, called weak-symmetry, in order to make possible reconstruction of mirror-symmetric 3D polygons. A novel algorithm is proposed to detect weak-symmetry in an input 2D polygon. The strategy of our algorithm requires the study of two kinds of weak-symmetry: the *internal weak-symmetry* and the *external weak-symmetry* (these notions are defined in Sec. 3.3). Detection of internal weak-symmetry is based on the calculus of intersection between at most n wedges (parts of the unit disc). The polygon P satisfies the internal weak-symmetry while this intersection is not empty. The external weak-symmetry of P is done in two steps. First, a candidate-directions set is computed, it contains every straight line that could make P weakly-symmetric. Second, the lines in the candidate-directions set are considered one after the other one and the external weak-symmetry of P is verified using a sweeping-line method.[12] Another algorithm is then used to reconstruct 3D polygons.

The rest of this paper is organized as follows: after the introduction of the required notation and of the related basic notions (Sec. 2), Sec. 3 presents the formal definition of the weak-symmetry problem and a main property that will be used to prove the principal results of this paper. Sections 4 and 5 study the weak-symmetry of a polygon, and present the related algorithms, accompanied by a complexity analysis. Section 6 presents the reconstruction step based on the

proposed algorithm. To improve readability of this paper the proofs of theorems and lemmas are presented in Appendices.

2. Preliminaries

Throughout this paper $P = (v_1, \ldots, v_n)$ will denote a *polygon* (without self-intersection) with *vertices* v_i $(i = 1, \ldots, n$ and $v_{n+1} := v_1)$ in the counterclockwise order, and with *edges* the segments $[v_i v_{i+1}]$.

A convex and a concave vertex. A vertex v_i of a polygon P is said to be a *convex vertex* if the unsigned angle $\angle v_{i-1} v_i v_{i+1}$ which is inside P is less than π, otherwise it is a *concave vertex*.

Among the key definitions of this paper are the following:

Type of a vertex. Let v_i be a vertex of the polygon P and ℓ an oriented line passing through v_i, then v_i is said to be of type **R** if and only if it satisfies one of the following properties:

(i) v_{i-1} and v_{i+1} are on the right side of ℓ.
(ii) v_{i-1} is on ℓ and v_{i+1} on the right side of ℓ.
(iii) v_{i+1} is on ℓ and v_{i-1} on the right side of ℓ.
(iv) If v_i has only one adjacent vertex then it is on the right side of ℓ.

We get the definition of a vertex of type **L** by replacing in the previous definition the term "right" by "left". Figure 2 shows the different cases and their correspondent type.

The ℓ-vertex notation. A vertex of type **R** or **L** with respect to a line ℓ is called an ℓ-vertex.

3. Weak-Symmetry of a Polygon with Respect to a Line

3.1. *Defining neighborhoods of points in polygons*

Let us denote with $]uw[$ the subset of the polygon P which is an open segment that does not contain its extremities u and w. Let r be a positive real number, then the neighborhood $V(x, r)$ of a point $x \in]v_i v_{i+1}[$ is the intersection set between the closed disc $b(x, r)$ of radius r and centered at x with P such that $b(x, r) \cap P \subset$

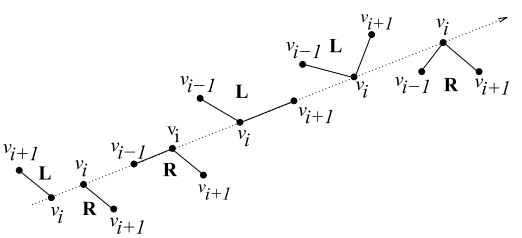


Fig. 2. The type of a vertex v_i.

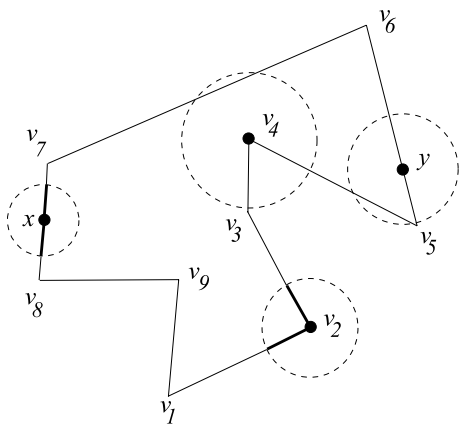


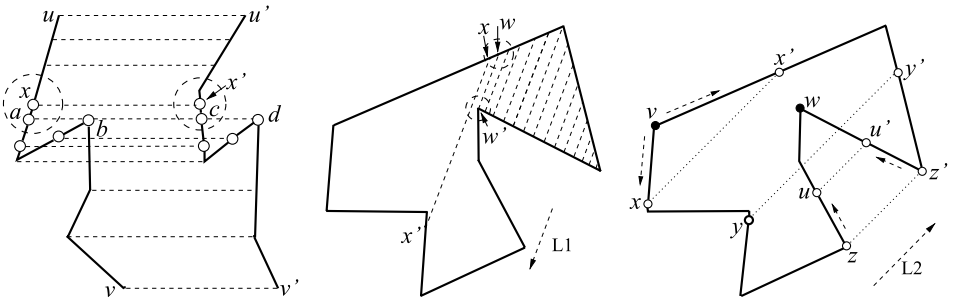
Fig. 3. Examples of neighborhoods of points on the polygon. The bold lines are neighborhoods of x and v_2.

$[v_i v_{i+1}]$. The neighborhood $V(v_i, r)$ of a vertex v_i is the set $b(v_i, r) \cap P$ that verifies $b(v_i, r) \cap P \subset [v_{i-1} v_i] \cup [v_i v_{i+1}]$. The neighborhoods $V(x, r)$ are not defined for every real positive r, however there exists $r_0 > 0$ sufficiently small so that for every $r < r_0$, $V(x, r)$ is well defined, that is, it satisfies the above inclusion constraint. In Fig. 3, the intersections of the discs centered at x and v_2 with the illustrated polygon are respectively included in $[v_7 v_8]$ and $[v_1 v_2] \cup [v_2 v_3]$, thus they define respectively neighborhoods of x and v_2. However, the intersections of the discs centered at $y \in [v_5 v_6]$ and v_4 with the polygon do not fall into the above inclusion constraint required by the neighborhood definition. For example, the dotted disc centered at v_4 cuts the edge $[v_6 v_7]$, so it does not satisfy the inclusion constraint that is, the intersection of the dotted disc centered at v_4 with the polygon should be included in $[v_3 v_4] \cup [v_4 v_5]$. Similarly, the dotted disc centered at y cuts the edge $[v_4 v_5]$. Thus, the inclusion constraint is not satisfied, since its intersection with the polygon is not included in $[v_5 v_6]$.

3.2. *The weak-symmetry notion*

Two polygonal segments P_1 and P_2 are *weakly-symmetric* with respect to ℓ if and only if there exists a mapping ϕ_ℓ from P_1 to P_2 such that:

(i) ϕ_ℓ is bijective, that is for all $x \in P_1$ there exists a unique $y \in P_2$ such that $y = \phi_\ell(x)$.
(ii) For all $x \in P_1$ the segment $[x\phi_\ell(x)]$ is parallel to ℓ or it is of zero length (i.e. $\phi_\ell(x) = x$).
(iii) ϕ_ℓ and ϕ_ℓ^{-1} are continuous. The continuity of ϕ_ℓ means that for all $x \in P_1$ there exists $\epsilon_0 > 0$ such that for all $\epsilon < \epsilon_0$, there exists $\delta > 0$ such that if $y \in V(x, \delta)$ then $\phi_\ell(y) \in V(\phi_\ell(x), \epsilon)$. The continuity of ϕ_ℓ^{-1} is obtained by replacing in the previous definition ϕ_ℓ by ϕ_ℓ^{-1}.



(a) (b) (c)

Fig. 4. (a) Two weakly-symmetric polygonal segments with respect to the horizontal. (b) The polygon is not weakly-symmetric with respect to the line $L1$. (c) The polygon is weakly-symmetric polygon with respect to the line $L2$. z, z', v, and w are the only $L2$-vertices of the illustrated polygon.

A polygon P is *weakly-symmetric with respect to* ℓ if and only if P can be divided in two polygonal segments P_1 and P_2 sharing their extremities and P_1 is weakly-symmetric to P_2 with respect to ℓ.

Example demonstrating cases of weak-symmetry presented in Figs. 4(a)–4(c). Figure 4(a) shows two polygonal segments which are weakly-symmetric to each other with respect to the horizontal line. The intersection of the horizontal (the dotted segments) with the polygonal segments gives the weakly-symmetric points. Each point must have a unique weakly symmetric point (bijection of the mapping). In the case of many intersections, we use the property of continuity. For example, consider the horizontal intersecting the two polygonal segments at a, b, c and d. Since a is in a neighborhood of x and c is the unique intersection-point which is in the neighborhood of x' (the point weakly-symmetric of x) then a is weakly-symmetric to c and therefore b is weakly-symmetric to d. In the example of Fig. 4(b), the polygon is not weakly-symmetric with respect to the line $L1$. The intersection points of the dotted segments with the polygon are the weakly-symmetric points. At the point w, the construction of the weakly-symmetric mapping is impossible, the point x is into the neighborhood of w, however x' is not in the neighborhood of w'. Consequently, the property of the mapping continuity is not satisfied. In the case of Fig. 4(c), the polygon is weakly-symmetric with respect to the line $L2$ because it can be divided onto two weakly-symmetric polygonal segments with respect to the line $L2$, these polygonal segments are P_1 of end points w, v and passing through x, and P_2 of end points v and w and passing through x'. The couples of the weakly-symmetric points are obtained by sweeping L over the polygon starting from the vertex v and ending at w. The intersection of L with the polygonal segments gives the weakly-symmetric points. For instance, the points weakly-symmetric to v, x, y, z, u and w are respectively $\phi_L(v) := v$, $\phi_L(x) := x'$, $\phi_L(y) := y'$, $\phi_L(z) := z'$, $\phi_L(u) := u'$ and $\phi_L(w) := w$.

To verify that a polygon is weakly symmetric with respect to a line ℓ, we need only to verify that the ℓ-vertices satisfy the key property stated in the next theorem, its proof is given in Appendix A.

Theorem 1. *If a polygon P is weakly-symmetric with respect to ℓ then each vertex of type* $\boldsymbol{R}$ (*respectively* $\boldsymbol{L}$) *is weakly-symmetric to a vertex of type* $\boldsymbol{R}$ (*respectively* $\boldsymbol{L}$).

3.3. *The internal and external weak-symmetry notions*

The study of the weak-symmetry notion needs to distinguish two kinds of weak-symmetry: The *Internal weak-symmetry* and the *External weak-symmetry*. In the rest of the paper, we denote them respectively as *I-weak symmetry* and *E-weak symmetry*. Before defining these notions, we introduce the following terminology. A line ℓ is called *internal-line* if for all $x \in P$, $[x\phi_\ell(x)]$ is included into the interior of P, that is $[x\phi_\ell(x)] \subset \mathrm{Int}(P) \cup P$ where $\mathrm{Int}(P)$ denotes the interior of P. Otherwise ℓ is called an *external-line*. We point out that the term external-line is an abuse of language since the segments $[x\phi_\ell(x)]$ are not everywhere out of the interior of P.

We say that a polygon P is *I-weakly symmetric* if there is at least one *internal-line* ℓ such that P is weakly-symmetric with respect to ℓ. A weakly-symmetric polygon which is not I-weakly symmetric is called *E-weakly-symmetric*.

The polygon of Fig. 5(a) is I-weakly symmetric because it is weakly-symmetric with respect to the internal-line $L1$. However, the polygon of Fig. 5(b) is not I-weakly symmetric but it is E-weakly symmetric with respect to the external-line L2. The detection of weak-symmetry employs two algorithms: the first deals with internal weak-symmetry, and it is presented in Sec. 4. The second algorithm detects the external weak-symmetry, and it is presented in Sec. 5.

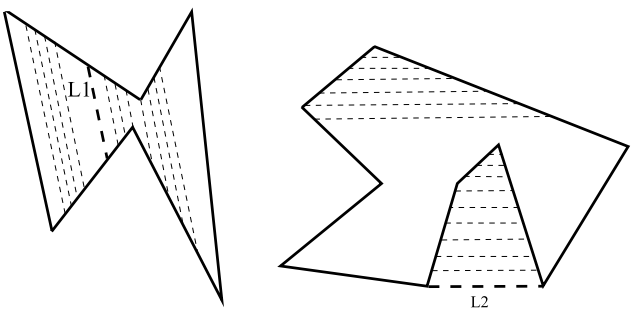


Fig. 5. (a) An I-weakly symmetric polygon. (b) This polygon is E-weakly symmetric. The extremities of each dotted segments are weakly-symmetric.

4. Detection of the Internal Weak-Symmetry of Polygons

4.1. *Internal weak-symmetry of convex polygons*

Theorem 2. *Every convex polygon P is I-weakly-symmetric for any line ℓ.*

Proof. Considering an arbitrary line ℓ, one must construct the weak-symmetry mapping ϕ_ℓ which satisfies the internal weak-symmetry constraint. Hereafter we define ϕ_ℓ. Since P is convex then we distinguish three cases:

(i) ℓ touches P at one vertex v: v is weakly-symmetric to itself, that is $\phi_\ell(v) := v$.
(ii) ℓ intersects P at exactly two points x and x': then x' is weakly-symmetric to x, thus $\phi_\ell(x) := x'$.
(iii) ℓ overlaps with exactly one edge $[v_1v_2]$ of P, then we pose $\phi_\ell(tv_1 + (1-t)\frac{v_1+v_2}{2}) := tv_2 + (1-t)\frac{v_1+v_2}{2}$, with $t \in [0,1]$.

The convexity of P ensures that for every $x \in P$, $[x\phi_\ell(x)] \subset \mathrm{Int}(P) \cup P$, which establishes that P is I-weakly symmetric. □

4.2. *Internal weak-symmetry of arbitrary polygons*

Consider the counterclockwise sense, for any vertex v, v^- and v^+ denote respectively the previous and the next vertex of v, and we say that a line separates v^- from v^+ if these two vertices are not strictly on the same side of this line. In this sub-section, we show that a polygon is I-weakly-symmetric if there exists a line ℓ such that for every concave vertex v of this polygon, the parallel line to ℓ passing through v separates v^+ from v^- (for an example see the two dotted lines of Fig. 6(a)).

Let $u_1, u_2, \ldots, u_j$ be the concave vertices of a polygon P. The *wedge* of u_k is the region outside P bounded by two half-lines starting at u_k and passing through u_k^- and u_k^+. In Fig. 6(a), the dashed region is the wedge of the vertex u_1. Let O be the origin of the employed coordinate system. After the translation of the wedge of u_k so that u_k coincides with O, we denote the resulting wedge $wedge_k$. $M_1(u_k)$ and $M_2(u_k)$ are the intersection points of the half-lines bounding $wedge_k$ with the unit circle centered at O.

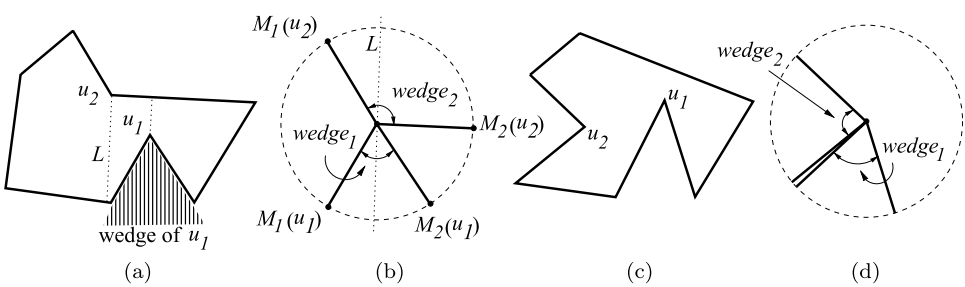


Fig. 6. (a) An I-weakly symmetric polygon. (b) The wedges of the polygon shown in (a). (c) A polygon which is not I-weakly symmetric. (d) The wedges of the polygon shown in (c).

Before introducing the next theorem, we point out that a line ℓ passing through the center O cuts $wedge_k$ if $M_1(u_k)$ and $M_2(u_k)$ are not on the same side of ℓ or ℓ passes through either $M_1(u_k)$ or $M_2(u_k)$. The next theorem states the main property used to detect the internal weak-symmetry of a polygon.

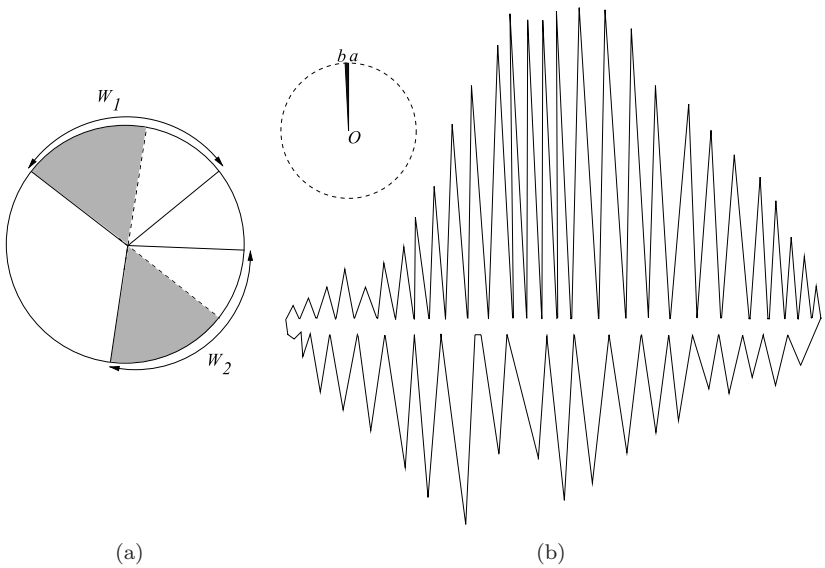
Theorem 3. *A polygon P is I-weakly-symmetric if and only if there exists a line ℓ, passing through O, which cuts all the wedges*: $wedge_1$, $wedge_2$, ..., $wedge_j$.

The proof is presented in Appendix A.

Examples. Figure 6(b) shows the wedges of the vertices u_1 and u_2 of the polygon in Fig. 6(a). The line L cuts the two wedges $wedge_1$ and $wedge_2$, therefore the polygon of Fig. 6(a) is I-weakly symmetric. Figure 6(d) shows the wedges of the polygon of Fig. 6(c): one can observe that there exists no line cutting both $wedge_1$ and $wedge_2$, thus the polygon of Fig. 6(c) is not I-weakly symmetric.

Let us call a *communication wedge between two wedges W_1 and W_2*, a wedge W such that each line passing through the origin O and cutting W cuts also W_1 and W_2 and vice versa. If W_1 and W_2 have a communication wedge, then there are two communication wedges between W_1 and W_2 (the gray wedges in Fig. 7(a)). They are symmetric with respect to the origin O. These two wedges have the same property: they identify all the lines cutting the two wedges W_1 and W_2. In the rest, we use only one of these communication wedges.

To detect the internal weak-symmetry of a polygon P, we compute iteratively the sequence of wedges W_k defined as follows: $W_1 = wedge_1$, W_k is the communication wedge between W_{k-1} and $wedge_k$, for all $k = 2, \ldots, j$. The wedge W_j groups



(a) (b)

Fig. 7. (a) Communication wedges between two wedges. (b) A polygon and the wedge returned by the algorithm IWSD.

all the lines making the polygon I-weakly symmetric. The algorithm IWSD (Internal Weak Symmetry Detection) summarizes this iterative process; see Algorithm 1. The time complexity of this algorithm is $O(n)$, which is established as follows: the loop in lines 4–6 iterates as many times as there are wedges. At each iteration, the communication wedge between two wedges is computed, which is done in constant time. Since the number of wedges does not exceed n then the time complexity of the algorithm IWSD is indeed linear.

Figure 7(b) presents the output of the algorithm IWSD applied on the illustrated polygon composed of 102 vertices. IWSD returns that the polygon is I-weakly symmetric with respect to every line passing through the illustrated wedge which is bounded by the segments $[Oa]$ and $[Ob]$, with $a = (-0.0026; 0.9999)$ and $b = (-0.0501; 0.9987)$.

```
1 Algorithm IWSD
  input  : A polygon P.
  output: A wedge W, each line cutting W makes P I-weakly-symmetric.
2 W ← wedge_1,
3 i ← 2,
4 while i ≤ j and W is not empty do
5 |   Update the wedge W : the new W is the wedge of communication between W
  |   and wedge_i,
6 end
7 return W,
```

Algorithm 1. Internal Weak-Symmetry Detection

5. Detection of External Weak-Symmetry

Let us turn to the case where the polygon is not I-weakly symmetric, and develop an algorithm detecting the E-weak symmetry. The algorithm EWSD (External Weak-Symmetry Detection) comprises two steps presented in the next sub-sections.

5.1. *Step* 1: *Computing the candidate directions set*

Given a polygon P, the goal is to compute a set called *candidate directions set*. This must be as small as possible, and contains all the lines, if any exists, which make P E-weakly symmetric. The algorithm EWSD (External Weakly Symmetric Detection) looks for all the lines making the input polygon E-weakly symmetric. At this point, some additional terminology must be introduced:

(a) A *concave-segment* of P is the segment e whose end-points are two concave vertices of P, which have the same type with respect to the line containing e.
(b) The definition of the *convex-segment* of P is obtained by replacing in the text of item (a) the term concave by convex.

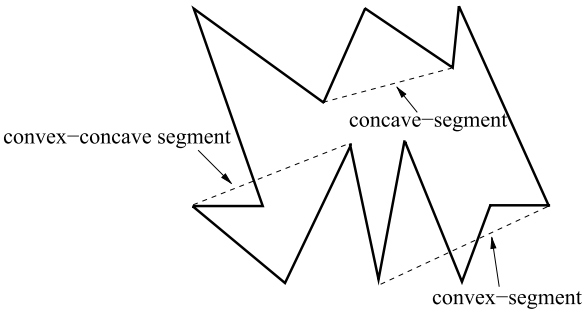


Fig. 8. Examples of convex, concave, and convex–concave segments in a polygon.

(c) A *convex–concave segment* is the segment e whose end-points are a convex and a concave vertex of P, and they have the same type (**R** or **L**) with respect to the line containing e.

Examples of convex, concave, and convex–concave segments are illustrated in Fig. 8.

We point out that a direction is represented by the unit director vector of a segment, in the following of the paper, the term segment means its directions. To define the candidate-directions set, the following sets are needed.

(d) $\mathcal{A}_0$ is the set of the convex–concave segments of P.
(e) $\mathcal{A}_1$ is the union of the three sets: $\mathcal{A}_0$ and the concave-segments of P which are not edges of P and the set of the edges of the convex hull of P which do not belong to P.
(f) $\mathcal{A}_2$ is the union of the two sets: $\mathcal{A}_0$ and the convex-segments of P which are not edges of P.

The sets $\mathcal{A}_1$ and $\mathcal{A}_2$ are used to define the candidate-directions set (Definition 1) because they contain all the lines, if they exist, that make P E-weakly symmetric (see the next theorem).

Theorem 4. *If P is E-weakly symmetric with respect to a line ℓ then ℓ is parallel to two segments, one belonging to the set $\mathcal{A}_1$ and the other belonging to $\mathcal{A}_2$.*

The proof of Theorem 4 is presented in Appendix B. Theorem 4 gives a way to filter the set of all the possible directions, that is the union of the convex edges, the concave segments and the convex–concave segments. The useless directions are those which do not belong to $\mathcal{A}_1 \cap \mathcal{A}_2$, they are removed. Therefore, based on this theorem, the candidate-directions set is defined as follows:

Definition 1. The set of the candidate directions $\mathcal{A}$ is the set of the directions ℓ which are parallel to a segment belonging to $\mathcal{A}_1$ and another segment belonging to $\mathcal{A}_2$.

Examples. In Fig. 9, the sets of the convex, concave as convex–concave segments are the following (the segments parallel to existing ones are not mentioned, they

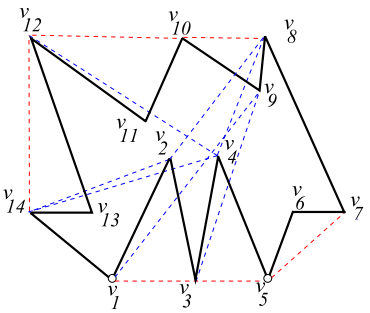


Fig. 9. (Color online) The dotted segments represent the candidate-direction set. The red segments are the convex hull edges and the blue ones are the convex–concave segments.

are redundant):

(i) The convex–concave segments are: $[v_1v_9]$, $[v_2v_8]$, $[v_2v_{14}]$, $[v_3v_9]$, $[v_4v_8]$, $[v_4v_{12}]$, $[v_4v_{14}]$.
(ii) The edges of the convex hull of P which do not belong to P are: $[v_1v_5]$, $[v_5v_7]$, $[v_{12}v_{14}]$.
(iii) The convex-segments which do not belong to the convex hull of P are: $[v_1v_7]$, $[v_3v_7]$, $[v_8v_{14}]$, $[v_{10}v_{14}]$.
(iv) The concave-segments are: $[v_9v_{11}]$, $[v_9v_{13}]$ and $[v_{11}v_{13}]$.

$\mathcal{A}_1$ is the union of the sets of items (i), (ii) and (iv), and the set $\mathcal{A}_2$ is the union of items (i), (ii) and (iii). According to Definition 1, the directions belonging to the candidate-directions set $\mathcal{A}$ are those belonging to (i) and (ii). The concave-segments are not added to $\mathcal{A}$ since they are not parallel to the convex-segments (the concave-segment $[v_2v_4]$ is not considered because it is parallel to the convex hull edge $[v_1v_5]$).

The function CDS computing the Candidate-Directions Set is summed up in Algorithm 2.

Complexity of the function CDS (*Algorithm* 2). The time complexity of the function CDS is at most $O(n^2)$. This is proved as follows: Line 2 computing the convex hull of P, costs $O(n)$.[13] The loops (line 5) iterate as many times as there are concave, convex and concave–convex segments: the number of these segments is $O(n^2)$ in the worst case. Therefore, the lines 5 to 25 cost $O(n^2)$, which gives the time-complexity of the function CDS.

5.2. *Step* 2: *Verifying the external weak-symmetry of a polygon*

The goal in this section is to find *all the lines* making the input polygon E-weakly-symmetric. We look for these lines in the candidate-directions set. That is, for each line of this set we verify that the polygon P is either E-weakly symmetric with respect to ℓ or not. To do this, we sweep ℓ over P. The sweep process consists of Steps A and B described below.

```
1  Function CDS
   input  : A polygon P.
   output: The candidate-directions set S.
2  Compute CH(P) the convex hull of P.
3  C ← the directions represented by the edges of CH(P) which do not belong to P.
4  i ← 0,
5  while i ≤ n do
6  |  j ← i + 2,
7  |  while j < n do
8  |  |  e ← [v_i v_j],
9  |  |  if e is convex–concave segment and e is not parallel to a direction of C
   |  |  then
10 |  |  |  add the direction e to C,
11 |  |  end
12 |  |  else
13 |  |  |  if e is a concave-segment and e is not parallel to a direction of CS then
14 |  |  |  |  add the direction e to CS,
15 |  |  |  end
16 |  |  |  else
17 |  |  |  |  if e is a convex-segment and e is not parallel to a direction of VS
   |  |  |  |  then
18 |  |  |  |  |  add the direction e to VS,
19 |  |  |  |  end
20 |  |  |  end
21 |  |  end
22 |  |  j ← j + 1,
23 |  end
24 |  i ← i + 1,
25 end
26 S ← (CS ∩ VS) ∪ C,
27 return S,
```

Algorithm 2. Candidate Directions Set

Step A: *Initialization* (*find two weakly-symmetric vertices*)

To start the sweep of a given line ℓ over the input polygon, two weakly-symmetric vertices are required. These vertices are defined by the next theorem. To simplify the presentation, without loss of generality, we consider the case where the line is the horizontal ℓ_o (with a rotation of P we can transform ℓ to ℓ_o).

Theorem 5. *If P is externally weakly-symmetric with respect to the horizontal then the vertex with the smallest ordinate and the smallest abscissa is weakly-symmetric to the vertex with the smallest ordinate and the largest abscissa.*

The proof of this theorem is presented in Appendix C. In the case of the polygon shown in Fig. 9, these two vertices are v_1 and v_5.

```
1  Function VEWSL
   input  : A polygon P, Two weakly-symmetric vertices u and v, A direction ℓ.
   output: ews a boolean, it takes true if P is E-weakly-symmetric otherwise its
           value is false.
2  u_p ← previous(u), v_n ← next(v),
3  u_n ← next(u), v_p ← previous(v),
4  b_1 ← u_p ≠ v_n and ([u_p v_n] is not an edge of P which is parallel to ℓ),
5  b_2 ← u_n ≠ v_p and ([u_n v_p] is not an edge of P which is parallel to ℓ),
6  ews ← true,
7  while (b_1 =true or b_2=true) and (ews=true) do
8      if (b_1 = true) then
9          c_1 ← (u_p and v_n belong to a line parallel to ℓ and have the same type
               (R or L) with respect to the line ℓ),
10         if (c_1 = true) then
11             u_p ← previous(u_p), v_n ← next(v_n),
12             b_1 ← u_p ≠ v_n and ([u_p v_n] is not an edge of P which is parallel to ℓ),
13         end
14         else
15             ews ← false,
16         end
17     end
18     if (b_2 = true) then
19         c_2 ← (u_n and v_p belong to a line parallel to ℓ and have the same type
               with respect to the line ℓ),
20         if (c_2 = true) and (ews = true) then
21             u_n ← next(u_n), v_p ← previous(v_p),
22             b_2 ← u_n ≠ v_p and ([u_n v_p] is not an edge of P which is parallel to ℓ),
23         end
24         else
25             ews ← false,
26         end
27     end
28 end
29 Return ews,
```

Algorithm 3. Verification of the External Weak-Symmetry with respect to a Line

Step B: *The sweep-line process*

Let v be an ℓ-vertex (for a definition see Sec. 2), and next(v) and previous(v) the next and the previous ℓ-vertex in the counterclockwise order. At Step A, two first weakly-symmetric vertices u and v with respect to ℓ are computed. Starting at u and v, ℓ is swept over P. The sweep stops at discrete "events", that is when ℓ hits ℓ-vertices. When u is identical to v the sweep stops iteratively at the couples defined by (next(u), previous(v)), otherwise the polygon is swept in two directions and thus we stop iteratively at (next(u), previous(v)) and (previous(u), next(v)). For each

```
1  Algorithm: EWSD
   input  : A polygon P
   output: The set L of the directions making P E-weakly symmetric.
2  S ← CDS (P),
3  Pose S = {ℓ_1, ..., ℓ_N},
4  L ← ∅,
5  for i=1 to N do
6  |  Compute the two first vertices u and v of P which are weakly symmetric
   |  with respect to ℓ_i (we use Theorem 5 to compute them),
7  |  ews ← VEWSL(P, u, v, ℓ_i),
8  |  if ews = true then
9  |  |  add ℓ_i to L,
10 |  end
11 end
12 Return the set L,
```

Algorithm 4. External Weak-Symmetry Detection

new couple, we verify that its vertices are weakly-symmetric (i.e. they must belong to the same line parallel to ℓ and must have the same type (**R** or **L**); this property is stated in Theorem 1). If once the property of Theorem 1 is not satisfied for a couple, we reject the direction ℓ, otherwise we return that P is E-weakly symmetric with respect to ℓ. The function VEWSL (Verification of the External Weak Symmetry with respect to a Line) summarizes the steps verifying the E-weak-symmetry of a polygon P with respect to a given line, see Algorithm 3. For simplicity this function is written for the case where the input first weakly-symmetric vertices u and v are different ($u \neq v$). The pseudo-code corresponding to the case $u = v$ can be obtained from this function by deleting the line 3, the variables b_2 and c_2, the line 5 and lines from 18 to 27. The algorithm EWSD summarizes the steps to detect the external weak-symmetry of a polygon, see Algorithm 4.

Example. Let us perform the lines 6 to 9 of the algorithm EWSD for the polygon of Fig. 9 with the direction ℓ_i equal to the horizontal line. The first computed weakly-symmetric vertices are v_1 and v_5 (line 6), the sweep process starts at v_1 and v_5. Using the function VEWSL (line 7), the horizontal hits respectively (v_{14}, v_6), (v_2, v_4), (v_{13}, v_7), (v_3, v_3), (v_{12}, v_8), these couples of vertices are weakly-symmetric. The next touched vertices are v_{11} and v_9 which are not weakly symmetric, since $[v_9 v_{11}]$ is not parallel to the horizontal line. Therefore, the function VEWSL returns false and the horizontal direction is rejected.

5.3. *Time complexity of the algorithm EWSD*

First we give the time complexity of the function VEWSL; see Algorithm 3. The loop (lines 7–28) iterates as many times as there are couples of vertices on the same line parallel to ℓ, thus the number of iterations is of order n in the worst

case. Since the tests of lines 8–17 and 18–27 are done in constant time then the function VEWSL costs $O(n)$ in the worst case. Now, we compute the worst-case complexity for Algorithm EWSD (Algorithm 4). It is established in Sec. 5.1 that the time complexity of the function CDS (called at line 2) is $O(n^2)$. Line 7 calls the function VEWSL, so it costs $O(n)$. The loop starting at line 5 iterates as many times as there are segments in the candidate-directions set S, that is $N = O(n^2)$ in the worst case. Therefore the worst complexity for the loop 5 is $O(n^3)$, which gives the time-complexity of Algorithm EWSD.

In practice, regarding the lines of the candidate-directions set which are not retained, the algorithm EWSD rejects them in few iterations, as it is demonstrated by the example in Table 1: the first column gives the candidate directions, for which the sweep starts at the vertices of column 2. Column 3 indicates the last touched vertices. For the whole weak-symmetry verification, the total number of visited vertices is given in column 4. We observe that for the rejected directions this number does not exceed 4 except for the direction $[v_1v_5]$. All the vertices are visited in the case of $[v_{12}v_{14}]$, since the polygon is weakly symmetric with respect to the line containing this segment.

5.4. *Experimental results*

The results of the algorithm EWSD on the polygons illustrated by Fig. 10 are presented in Table 2. They show the behavior of the algorithm on polygons with different shapes. The first row gives the number of vertices of each polygon, the second row shows the number of convex segments, these values include the number of the convex hull edges of the polygons. The third and fourth rows give respectively the numbers of concave and convex–concave segments. The number

Table 1. Related to Fig. 9. NVV: the number of the visited vertices.

Directions	The sweep starts at:	The sweep stops at:	NVV
$[v_1v_9]$, $[v_2v_8]$, $[v_4v_8]$	v_7	(v_6, v_8)	2
$[v_3v_9]$	v_7	(v_4, v_8)	4
$[v_2v_{14}]$	v_5	(v_6, v_4)	2
$[v_5v_7]$	(v_5, v_7)	(v_4, v_8)	3
$[v_1v_5]$	(v_1, v_5)	(v_9, v_{11})	11
$[v_{12}v_{14}]$	(v_{12}, v_{14})	v_7	14

Table 2. Outputs of the algorithm EWSD on the polygons of Fig. 10.

	Fig. 10(a)	Fig. 10(b)	Fig. 10(c)	Fig. 10(d)	Fig. 10(e)
Number of vertices	34	51	114	206	631
Convex segments	69	72	3	652	7174
Concave segments	87	100	1317	556	6815
Convex–concave segments	5	115	0	1101	13445
Candidate-directions	62	179	2	1611	18831
CPU time (sec)	0	0	0	0.03	0.81

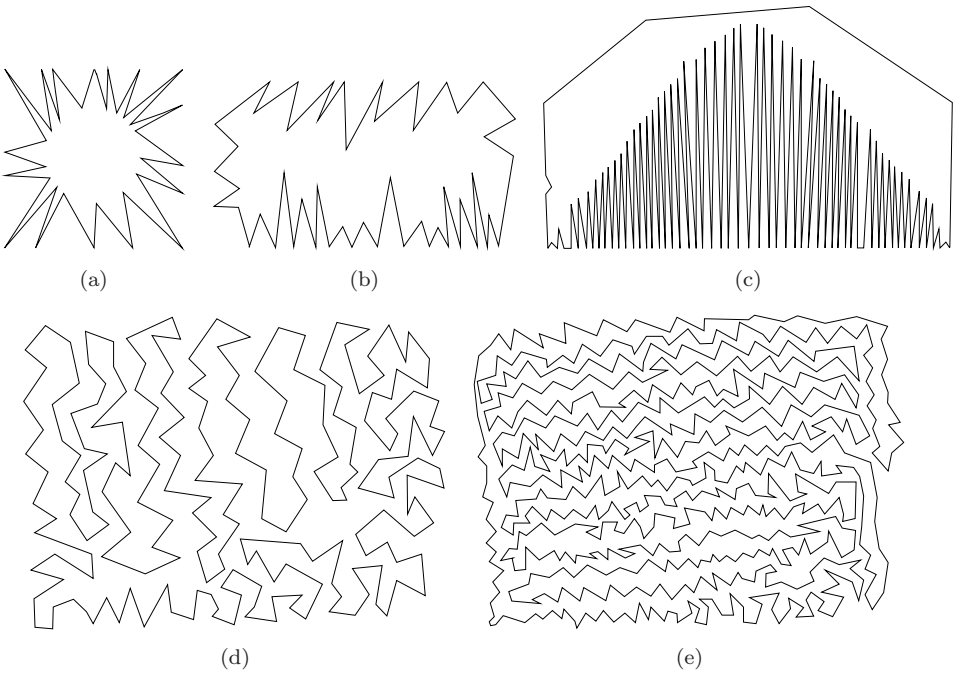


Fig. 10. Related to Table 2.

of candidate-directions that could make the polygon E-weakly-symmetric is given in the fifth row. For the polygon of Fig. 10(c), the algorithm is efficient since it selects only two candidate-directions. Compared to an algorithm which verifies the weak-symmetry for all the possible directions (all the segments connecting two vertices of a polygon), our method filters out many of the directions to consider for identifying the E-weak-symmetry. In fact, the numbers of the candidate-directions are significantly smaller than the following numbers of directions considered by a brute-force algorithm: 486, 1125, 4846, 20 910 and 198 135 corresponding respectively to Figs. 10(a)–10(e). The last row presents the running time (on 2.67 GHZ with 4 GB RAM), we observe that the algorithm is fast, it takes less than 1 s even for polygons with large number of vertices and having complex shapes.

6. 3D Reconstruction Using the Weak-Symmetry Correspondence

Once the weak-symmetry correspondence among the vertices of the polygon has been calculated, the next step is to compute the 3D positions of these vertices. Let v a vertex of coordinates (x, y, z) and v' a vertex of coordinates (x', y', z'). v' is the mirror-symmetric of v. We assume that v and v' do not have same coordinates. Let M be the symmetric plane whose normal vector $\vec{N}$ is a unit vector of coordinates (x_n, y_n, z_n). The two vertices are orthogonally projected onto the plane $(z = 0)$. Thus, the 3D reconstruction comes down to computing the z-coordinates of the two

vertices. As explained in Ref. 14m if the coordinates (x_n, y_n, z_n) with $z_n \neq 0$, on the normal vector of M is known, the z-coordinates of the two vertices are given by Eqs. (1) and (2):

$$z = -\frac{1}{2}\left(\frac{x_n(x'+x)}{z_n} + \frac{y_n(y'+y)}{z_n} + \frac{z_n(y'-y)}{y_n}\right). \tag{1}$$

$$z' = -\frac{1}{2}\left(\frac{x_n(x'+x)}{z_n} + \frac{y_n(y'+y)}{z_n} - \frac{z_n(y'-y)}{y_n}\right). \tag{2}$$

The z-coordinates are also given by Eqs. (3) and (4):

$$z = -\frac{1}{2}\left(\frac{x_n(x'+x)}{z_n} + \frac{y_n(y'+y)}{z_n} + \frac{z_n(x'-x)}{x_n}\right). \tag{3}$$

$$z' = -\frac{1}{2}\left(\frac{x_n(x'+x)}{z_n} + \frac{y_n(y'+y)}{z_n} - \frac{z_n(x'-x)}{x_n}\right). \tag{4}$$

Given that the value of z is given by Eqs. (1) and (3), the coordinates of v and v' satisfy the following equality:

$$\frac{(x'-x)}{(y'-y)} = \frac{x_n}{y_n}. \tag{5}$$

Equation (5) gives the relation between the coordinates x_n and y_n of the normal vector $\vec{N}$. The other coordinate z_n must be chosen between 0 and 1. Since $\vec{N}$ is a unit vector and (x_n, y_n) satisfies Eq. (5) then the normal vector $\vec{N}$ is determined and thus the symmetric-plane of the 3D polygon can be computed. By varying the values of z_n within the interval $(0, 1]$, the family of the reconstructed 3D polygons are identical under scaling change of factor related to z_n. In this paper, we do not develop a method to select automatically the value of z_n for reconstructing the curve at the appropriate scale. For the examples of Fig. 11 z_n is set to 0.5. One possibility to define the z_n value automatically would be to choose it in a way that the compactness of the reconstructed curve is maximized. The compactness could be defined as V^2/S^3, V being the volume of the bounding box of the reconstructed curve and S its surface area. Li *et al.*[3] have shown that human beings interpret sketches in a way to maximize the compactness of the reconstructed shape. We leave this as future work. Note that Eq. (5) implies that one of the two coordinates x_n and y_n must be different from 0. If x_n is equal to 0, the computation of the z-coordinates of v and v' is done with Eqs. (1) and (2). If y_n is equal to zero, Eqs. (3) and (4) should be used instead. If x_n and y_n are both different from 0 then one may employ any or all of Eqs. (1)–(4).

Figure 11 shows two examples of 3D reconstruction using our algorithms IWSD and EWSD. We have computed the line that makes all the polygons in each (Figs. 11(a) and 11(c)) weakly-symmetric. The tangent vector of these two lines are (0.655; 0.755) and (0.309; 0.951) for Figs. 11(a) and 11(c) respectively. The process of computing the weak-symmetry of a set of polygonal curves is as follows.

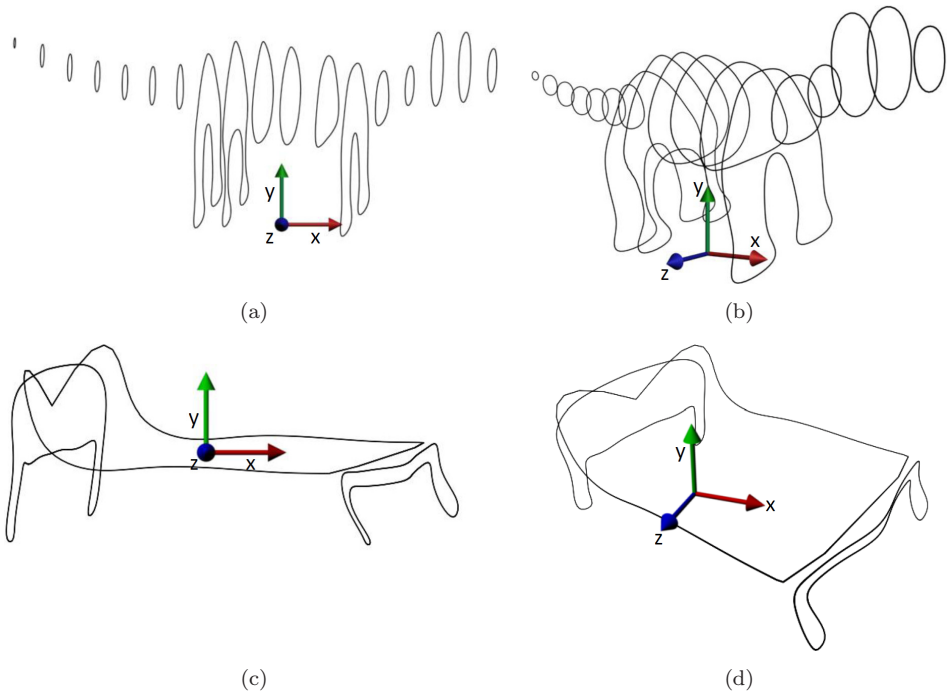


Fig. 11. Drawings composed of 2D polygons (a) and (c) and the corresponding 3D reconstruction (b) and (d).

First, we compute all possible directions of the weak-symmetry for each curve individually. For convex curves, the weak-symmetry can be along any direction. For nonconvex curves, there are usually one or two possible directions. Next, we find if there exists one direction of the weak-symmetry which is common to all the curves. If so, this implies that these curves are the projection of 3D polygonal curves which are symmetric with respect to the same plane. We compute the 3D positions of these curves using the common direction and Eqs. (1)–(4).

We point out that when all the curves are weakly-symmetric with respect to more than one direction, this involves that different 3D polygonal curves can be reconstructed. Each direction of weak-symmetry defines a plane such that the input curves are the projection of 3D curves which are symmetric with respect to this plane. Similarly to the computation of the z_n value, we could use the compactness criteria to find the direction of reconstruction automatically. One way to do it would be to enumerate all the possible directions and choose the one which maximize the compactness of the reconstructed curves. We leave it as future work.

7. Conclusion

Detecting the weak-symmetry of a planar hand-sketched polygonal curve is a key step in the process of reconstructing mirror-symmetric nonplanar 3D polygonal

curves. In this paper, weak-symmetry has been formally defined, and a method to identify it has been developed. The whole time complexity of the "weak-symmetry identification" algorithm is $O(n^3)$ in the worst case. Our future work will be to extend this method to the problem of detecting the weak-symmetry of polygons with self-intersections.

Appendix A. Proof of Theorem 1 and Theorem 3

A.1 *Proof of Theorem* 1

Consider a vertex v which is of type **R** with respect to a fixed line ℓ, and suppose that the point v' which is weakly symmetric to v is not of type **R**. Then, we have four possible situations illustrated in Fig. 12. In all cases, v is weakly-symmetric to a point v' which is not a vertex of type **R**. The line touching v and v' obtained by a slight move of ℓ, intersects P at the point a in a neighborhood of v'. However, this line does not intersect P in a neighborhood of v. That is $\phi_\ell(a)$ is not in a neighborhood of v which contradicts the continuity of ϕ_ℓ and thus contradicts the weak-symmetry of P.

A.2 *Proof of Theorem* 3

Let us prove the first implication. Let P be an I-weakly-symmetric polygon and u_k be a concave vertex of P. We will prove that ℓ cuts $wedge_k$, for every $k = 1, \ldots, j$. To derive a contradiction, we suppose that ℓ passing through u_k does not cut the wedge of u_k, which implies that u_k is either of type **R** or of type **L**, say that it is of type **R**. Theorem 1 ensures that u_k is either weakly-symmetric to itself or to a vertex v of type **R** (see the example of Figs. 13(a) and 13(b)). Let us move slightly the line ℓ touching u_k. In the case where u_k is weakly-symmetric to itself, the obtained line intersects P at two points a and b in a neighborhood of u_k (Fig. 13(a)). The

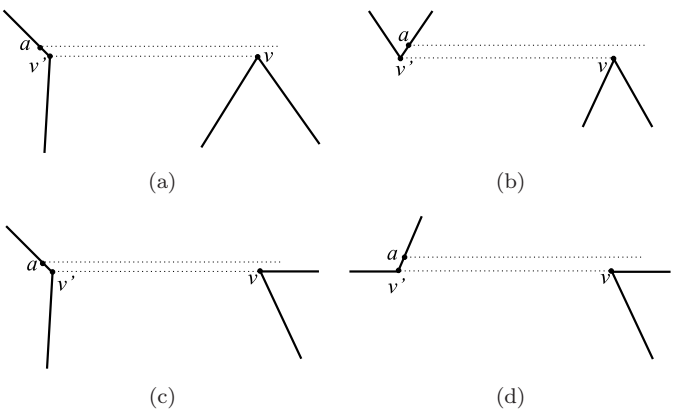


Fig. 12. Proof of Theorem 1: Illustration of the possible situations where the weakly-symmetric vertices v and v' have not the same type.

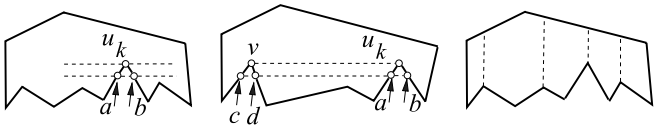


Fig. 13. Related to the proof of Theorem 3.

continuity of the mapping ϕ_ℓ implies that a is weakly-symmetric to b ($\phi_\ell(a) = b$) and since u_k is concave then the segment $[ab]$ is not included in the interior of P. This contradicts the internal weak-symmetry of P. Let us turn to the case where u_k is weakly-symmetric to v, the obtained line, with a slight move of ℓ, intersects P at the points a and b in a neighborhood of u_k and at two points c and d in a neighborhood of v (see the example of Fig. 13(b)). By reason of continuity of ϕ_ℓ the weakly-symmetric points to a and to b are c and d ($\{\phi_\ell(a), \phi_\ell(b)\} \subset \{c, d\}$). Because u_k is concave, one of the segments $[a\phi_\ell(a)]$ or $[b\phi_\ell(b)]$ is not included in $\mathrm{Int}(P)$. In the example of Fig. 13(b) the segment $[b\phi_\ell(b)]$ is outside P: neither $[bc]$ nor $[bd]$ is included in $\mathrm{Int}(P)$. This contradicts the fact P is I-weakly symmetric. Therefore ℓ must cut $wedge_k$ for any concave vertex u_k.

Conversely, let us prove that if there exists a line ℓ that cuts all the wedges $wedge_k$, $k = 1, \ldots, j$ then P is I-weakly symmetric with respect to ℓ. Since the line ℓ cuts the wedge of each concave vertex u_k, then u_k is neither of type **R** nor of type **L**. Let us divide P into convex polygons using the lines parallel to ℓ and passing through the concave vertices u_k (see Fig. 13(c)). Theorem 2 ensures that the convex polygons are I-weakly symmetric with respect to ℓ thus P is also I-weakly-symmetric with respect to ℓ. This ends the proof.

Appendix B. Proof of Theorem 4

To prove Theorem 4, we need to prove lemmas 1–4. All lemmas between 1 and 4, stated immediately below, consider that the *polygon P is E-weakly symmetric with respect to a given line ℓ.*

Lemma 1. *The line ℓ is parallel to an edge of the convex hull of P or to a segment connecting two vertices of P having the same type* (**R** *or* **L**) *with respect to ℓ.*

Proof. Let P be an external weakly-symmetric with respect to a line ℓ, we should prove that ℓ is parallel to an edge of the convex hull of P or to a segment connecting two vertices having the same type (**R** or **L**) with respect to ℓ. Let us sweep ℓ over P starting outside the convex hull of P, then ℓ will hit:

(i) at least two vertices of P which have the same type (the line A in Fig. 14), or
(ii) an edge of P (the line B in Fig. 14), or
(iii) a vertex of P (the line C in Fig. 14).

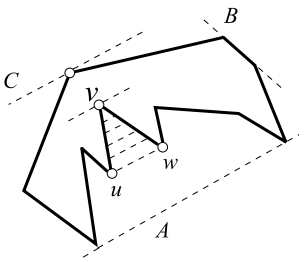


Fig. 14. Related to the proof of Lemma 1.

In the cases (i) and (ii), ℓ is parallel to an edge of $CH(P)$ which achieves the proof. It remains the case (iii), let us suppose that ℓ hits a vertex of P (case of the line C). Starting at this vertex, we sweep ℓ over P until it hits a vertex v of type **R** or **L**, say **R**, see Fig. 14. This vertex exists, otherwise P is I-weakly symmetric with respect to ℓ, which contradicts the definition of the external weak-symmetry of P. As v is of type **R** then Theorem 1 ensures that there is only two cases:

(a) v is weakly-symmetric to a vertex v' of type **R**, or
(b) the vertex v is weakly-symmetric to itself.

In the case (a) the proof is achieved: $[vv']$ is the segment we are looking for. It remains the case (b): let u and w be respectively the previous and the next ℓ-vertices from v. Consider the sweep of the line ℓ starting at v and ending at u and w (see the dotted segments between v and $[uw]$, Fig. 14). By reason of the continuity of the weak-symmetry mapping, the intersection points of the dotted lines with the $[uv]$ and $[uw]$ are weakly-symmetric. We deduce that u must be weakly-symmetric to w and thus they have the same type (Theorem 1). Consequently, the line ℓ is parallel to a segment connecting two vertices of the same type, which ends the proof. □

The next lemma is introduced to prove Lemma 3.

Lemma 2. *Let $[uv]$ be an edge of P parallel to ℓ. If u is not weakly-symmetric to v then $[uv]$ is weakly-symmetric to an edge of P aligned with $[uv]$. Otherwise the half-edge $[u\frac{u+v}{2}]$ is weakly-symmetric to $[v\frac{v+u}{2}]$.*

Proof. Consider an edge $[uv]$ parallel to ℓ. First, let us study the case where $\phi_\ell(u) \neq v$. As $[u\phi_\ell(u)]$ and $[v\phi_\ell(v)]$ are parallel to ℓ then u, v, $\phi_\ell(u)$ and $\phi_\ell(v)$ are on the same line parallel to ℓ. The continuity of ϕ_ℓ and ϕ_ℓ^{-1} ensure that $[\phi_\ell(u)\phi_\ell(v)]$ is necessarily an edge of P. Now, let us turn to the case where v is weakly-symmetric to u, i.e. $\phi_\ell(u) = v$. The continuity of the mapping ϕ_ℓ ensures that each point on $[uv]$ is weakly-symmetric to a point belonging to the same edge $[uv]$. For each point on the edge $[uv]$, the weak-symmetry mapping can be defined as follows: $\phi_\ell(tv + (1-t)\frac{u+v}{2}) := tu + (1-t)\frac{u+v}{2}$, with $t \in [0,1]$. □

Lemma 3. *If ℓ is parallel to an edge e of P then e is parallel to a segment $e' \notin P$ which is convex or concave or convex–concave.*

Proof. Consider an edge $e = [uv] \in P$ parallel to the direction ℓ, we should prove that e is parallel to a concave segment or to a convex segment or to a convex–concave one (cf. Sec. 5.1 items (a), (b) and (c)).

Based on Lemma 2, there are only two cases:

(i) $e = [uv]$ is weakly-symmetric to an edge $e' = [u'v'] \in P$ where e' is different than e, or
(ii) e is weakly-symmetric to itself.

Consider the case (i) which is illustrated by Fig. 15(a), the vertices u and v are respectively weakly-symmetric to u' and v'. Since the vertices v and v' have the same type and $[vv'] \notin P$ then $[vv']$ is the segment parallel to e, which we are looking for (in the example of Fig. 15(a), it is a convex segment). This ends the proof for this case. Let us turn to the case (ii) where the edge e is weakly-symmetric to itself, this case is shown in Fig. 15(b), starting at e, we sweep ℓ over P as illustrated by the dotted segments, the intersection points of these lines with the polygon are weakly-symmetric (consequence of the continuity of the weak-symmetry mapping). The line ℓ must meet two vertices u' and v' which have the same type. Furthermore $[u'v']$ cannot belong to P, otherwise P will be I-weakly-symmetric which will contradict the fact P is E-weakly symmetric. This ends the proof of the case (ii) and thus the proof of the lemma. □

Lemma 4. (i) *If the line ℓ is parallel to a concave-segment e of P which is not an edge of P, then there exists a convex-segment e' of P parallel to ℓ and in addition e' is not an edge of P.*

(ii) *If the line ℓ is parallel to a convex-segment e of P which is neither an edge of P nor an edge of the convex hull of P, then there exists a concave-segment e' of P parallel to ℓ and in addition e' is not an edge of P.*

Proof. (i) To simplify the proof of the implication, we refer to Fig. 16. Let $e = [uv] \notin P$ be a concave segment and u is weakly-symmetric to v. Starting

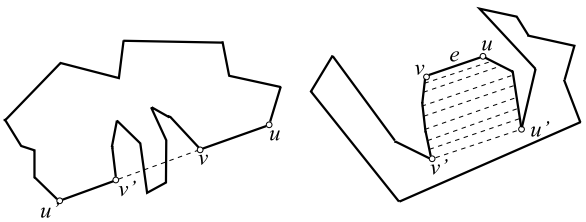


Fig. 15. (a) and (b) are related to the proof of Lemma 3.

at u and v, we sweep the line containing e over P as illustrated by the dotted segments, the intersection points (empty small discs) of these lines with the polygon are weakly-symmetric (consequence of continuity of the weak-symmetry mapping). The sweeping stops when the line hits two convex vertices u' and v' having the same type (**R** or **L**). These vertices u' and v' must exist. Let us prove it: since u and v are concave then the line containing e divides the polygon into two parts (the polygon cannot be on the same side of this line because a concave segment is not an edge of the convex hull of P). Starting at e, we sweep the line containing e over the polygon, the sweeping takes the direction opposite to w, that is, we move away from w (cf. Fig. 16). This sweeping must reach the vertex w, which is possible only in the case of existence of two convex vertices to reverse the sweeping direction and move closer to w. The segment connecting these two convex vertices is the segment we are looking for. This ends the proof of this property.

(ii) The proof of the property presented in Lemma 4(ii) is similar to the proof of the property (i). We replace in the previous proof "concave" by "convex" and "convex" by "concave". □

B.1 *Proof of Theorem* 4

The proof of this theorem is a direct consequence of Lemma 1, 3 and 4. Lemma 1 ensures that the line ℓ is parallel to a segment e_1 which is a convex-segment or a concave-segment or a convex–concave segment. Lemma 3 shows that it is useless to consider the edges of P, thus $e_1 \notin P$. In the case where e_1 is a convex hull edge or convex–concave segment, the property is satisfied: $e_1 \in \mathcal{A}_1$ and $e_1 \in \mathcal{A}_2$. This ends the proof for this case. It remains the case where e_1 is a concave-segment or a convex-segment which is not edge of the convex hull of P. Consider the case where e_1 is a concave-segment, thus $e_1 \in \mathcal{A}_1$. Lemma 3 ensures that there is a convex-segment e_2, (thus $e_2 \in \mathcal{A}_2$) and parallel to the segment e_1. The same reasoning can be used for the case where e_1 is a convex-segment.

Appendix C. Proof of Theorem 5

Consider the vertices $u_1 < u_2 < \cdots < u_r$ having the smallest ordinate and sorted according to their increasing abscissa. To prove that the lowest leftmost vertex u_1

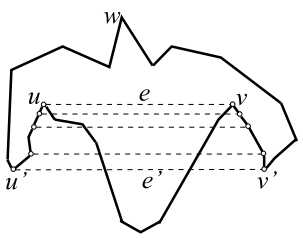


Fig. 16. Related to the proof of Lemma 4.

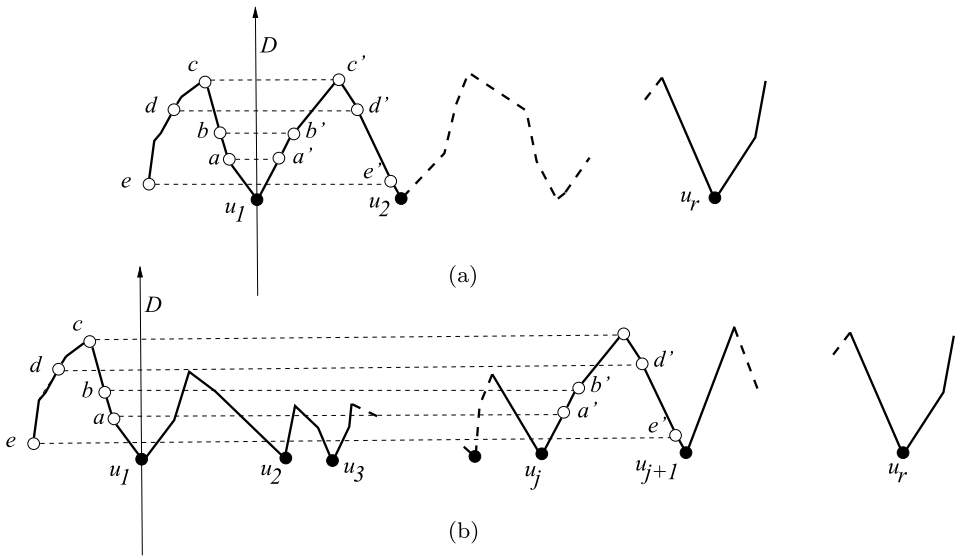


Fig. 17. Related to the proof of Theorem 5.

is weakly-symmetric to the lowest right most vertex u_r, we refer to Fig. 17(a). Let u'_1 be the vertex weakly-symmetric to u_1 and suppose that $u_r \neq u'_1$. Theorem 1 ensures that u_1 and u'_1 have the same type, which implies that $u'_1 \in \{u_1, \ldots, u_r\}$. Therefore, we have two possible cases $u'_1 = u_1$ or $u'_1 = u_j$ with $1 < j < r$.

Consider the case $u'_1 = u_1$ and try to derive a contradiction. Starting at u_1, we sweep the horizontal over P, the direction of the sweeping is illustrated by the dotted segments (cf. Fig. 17(a)): the order is $[aa']$, $[bb']$, $[cc']$, $[dd']$ and $[ee']$, we stop the sweep until we reach the vertex u_2. The continuity of the weak-symmetry mapping, ensures that the vertex weakly-symmetric to u_2 must be in the left side of the vertical line D passing through u_1, this contradicts the fact that u_1 is the lowest leftmost vertex.

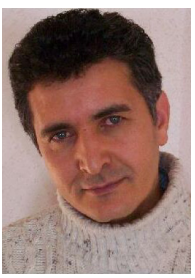
Now let us turn to the second case and suppose that $u'_1 = u_j$ with $1 < j < r$. Since u_1 and u_j are weakly-symmetric then these vertices divide P into two weakly-symmetric polygonal segments with respect to the horizontal. The first one starts at u_1 and ends at u_j and composed of the vertices $u_2, u_3, \ldots, u_r$. Here, we are interested by the other polygonal segment. Starting at u_1 and u_j, we sweep the horizontal over this polygonal segment, as shown by Fig. 17(b). The horizontal passes consecutively through $[aa']$, $[bb']$, $[cc']$, $[dd']$ and $[ee']$, we continue the sweeping until we reach u_{j+1}. As illustrated by Fig. 17(b), the vertex weakly-symmetric to u_{j+1} must be in the left side of the vertical line D, this contradicts the fact that u_1 is the lowest leftmost vertex. Consequently u_1 must be weakly-symmetric to u_r.

Acknowledgments

The research of N. Sapidis has been co-financed by the European Union (European Social Fund — ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) — Research Funding Program: THALIS.

References

1. L. Olsen, F. F. Samavati, M. C. Sousa and J. A. Jorge, "Sketch-based modeling: A survey," *Comput. Graph.* **33**(1), 85–103 (2009).
2. S. U. Cheon and S. Han, "A template-based reconstruction of plane symmetric 3D models from freehand sketches," *Computer-Aided Design* **40**(9), 975–986 (2008).
3. Y. Li, Z. Pizlo and R. M. Steinman, "A computational model that recovers the 3D shape of an object from a single 2D retinal representation," *Vision Res.* **49**(9), 979–991 (2009).
4. N. Jiang, P. Tan and L. F. Cheong, "Symmetric architecture modeling with a single image," *ACM Trans. Graph.* **28**(5), 1–8 (2009).
5. O. Tolba, J. Dorsey and L. McMillan, "A projective drawing system," in *Proc. Symp. Interactive 3D Graphics* (2001), pp. 25–34.
6. J. Cohen, L. Markosian, R. Zeleznik, J. Hughes and R. Barzel, "An interface for sketching 3D curves," in *Proc. Symp. Interactive 3D Graphics* (1999), pp. 17–21.
7. T. Kanade, "Recovery of the three-dimensional shape of an object from a single view," *Artif. Intell.* **17**, 409–460 (1981).
8. S. Posch, "Detecting skewed symmetries," *11th IAPR Int. Conf. Pattern Recogn.* **3**, 602–606 (1992).
9. D. Shen, H. H.-S. Ip and E. K. Teoh, "Robust detection of skewed symmetries by combining local and semi-local affine invariants," *Pattern Recogn.* **34**(7), 1417–1428 (2001).
10. D. Shen, H. H.-S. Ip and E. K. Teoh, "Robust detection of skewed symmetries," *Int. Conf. Pattern Recogn.* 7022–7025 (2000).
11. Y. Raymond KK, "A Hough transform technique for the detection of reflectional symmetry and skew-symmetry," *Pattern Recogn. Lett.* **21**(2), 117–130 (2000).
12. A. Okabe, B. Boots and K. Sugihara, *Spatial Tessellations*: *Concepts and Applications of Voronoi Diagrams* (John Wiley and Sons, New York, 1992).
13. A. Melkman, "On-line construction of the convex hull of a simple polygon," *Inform. Process. Lett.* **25**, 11–12 (1987).
14. F. Cordier and H. Seo, "Free-form sketching of self-occluding objects," *IEEE Comput. Graph. Appl.*, special issue on Sketching **27**, 50–59 (2007).
15. F. Cordier, H. Seo, J. Park and J. Y. Noh, "Sketching of mirror-symmetric shapes," *IEEE Transact. on Vis. and Comput. Graphics* **17**(11), 1650–1662 (2011).



Mahmoud Melkemi received a PhD in applied mathematics from the University of Grenoble 1, France. He is a researcher of LMIA(Laboratoire de Mathmatiques Informatique et Applications) at Haute Alsace University, Mulhouse, France. His research works are focused on pattern recognition, computer graphics, and computational geometry with applications in ad hoc wireless communication and sensor networks.

Frederic Cordier received his PhD degree in Computer Graphics from the University of Geneva, Switzerland in 2004. From 2004 to 2007, he worked as a Post-Doctoral researcher in the Department of Computer Science at KAIST. He joined the Upper Alsace University in 2007 as an Associate Professor. His main research interests include 3D modeling and texturing, human computer interaction and physics-based simulation.

Nickolas S. Sapidis is a Professor with the Mechanical Engineering Department of the University of Western Macedonia (Greece). He has been a faculty member with the University of the Aegean (Greece), and he has also taught at the National Technical University of Athens, the University of Athens, and the Polytechnic University of Catalunya (Spain). N. Sapidis has been pursuing research on "Mechanical Design", "Computer-Aided Design (CAD)", "Geometric and Solid Modeling", and "Computer Graphics". He is the author of more than 70 papers on these subjects. His industrial experience includes the General Motors R&D Center (USA) and the Marine Technology Development Co. (Greece). N. Sapidis is on the editorial board of several international scientific journals including *Computer-Aided Design*, *Virtual and Physical Prototyping*, *International Journal of Product Lifecycle Management*, *Computer-Aided Design and Applications*, *Mathematical Problems in Engineering*, and *International Journal of Computer-Aided Engineering and Technology*. N. Sapidis has served on the Program Committee of more than 60 international conferences, and was an "invited Keynote Speaker" for seven international conferences.