

# Detecting and Quantifying Envelope Singularities in the Plane\*

Hüseyin Erdim and Horea T. Ilies<sup>†</sup>  
Computational Design Laboratory  
University of Connecticut<sup>‡</sup>  
Email: *herdim,ilies@engr.uconn.edu*

## Abstract

The mathematical envelopes to families of both rigid and non-rigid moving shapes play a fundamental role in a variety of problems from very diverse application domains, from engineering design and manufacturing to computer graphics and computer assisted surgery. Geometric singularities in these envelopes are known to induce malfunctions or unintended system behavior, and the corresponding theoretical and computational difficulties induced by these singularities are not only massive, but also well documented.

We describe a new approach to detect and quantify the envelope singularities induced by 2-dimensional shapes of arbitrary complexity moving according to general non-periodic and non-singular planar affine motions. Our approach, which *does not require any envelope computations*, is reframing the problem in terms of “fold points” and “fold regions” in the neighborhood of geometric singularities, and we show that the existence of these fold points is a necessary condition for the existence of singularities. We establish a mathematically well defined duality between the 2-dimensional Euclidean space in which the motion takes place and a  $2 + 1$  spacetime domain. Based on this duality, we recast the problem of *detecting* and *quantifying* geometric singularities into inherently parallel tests against the *original geometric representation* in the 2-dimensional Euclidean space. We conclude by discussing the significance of our results, and the extension of our approach to 3-dimensional moving shapes.

**Keywords:** singularities, envelopes, undercutting, fold points, sweep, higher pairs.

## 1 Motivation

Sweeping an object through space is one of the fundamental operations in geometric modeling [31]. Furthermore, many practical problems from very diverse fields ranging from engineering design and manufacturing to robotics, computer graphics and computer assisted surgery can be formulated in terms of sweeps [42, 14, 40, 39, 25, 33]. The boundary of a set swept by an object in motion is expressed mathematically as the envelope to the family of shapes defined by the moving object [5]. These envelopes, which are defined as solutions to specific differential equations, are known to exhibit geometric singularities. In practical terms, these envelope singularities induce malfunctions

---

\*This paper has appeared in Computer Aided Design, Vol. 39, no. 10, pp 829-840, October 2007.

<sup>†</sup>Corresponding author.

<sup>‡</sup>Department of Mechanical Engineering, Storrs, CT 06269-3139

or unintended system behavior in all the corresponding applications<sup>1</sup>. However, the theoretical and computational difficulties induced by these singularities are not only massive, but also well documented.

Consider a simple cam-follower mechanism used to transform a rotation with constant angular velocity of a shaft into a relatively complex output motion, such as that illustrated in Figure 1. Such a design problem is typically formulated in terms of the set swept by the moving follower relative to the cam [40, 47, 17]. The boundary of such a cam is a subset of the envelopes of the moving follower (in this case a 2D disk). Though, for a given follower as well as cam and follower motions, the envelopes of the moving follower will not always be in contact with the follower. In other words, a cam that would move the follower according to the prescribed motion may not exist. This case is illustrated in Figure 1, where the follower is moving relative to the cam such that its center moves along the specified trajectory. In this example, loss of contact between the follower and the resulting external cam would occur twice during a complete cycle of the relative motion. Observe that the envelopes corresponding to the loss of contact have singular points, which is discussed in more detail in section 2. This phenomenon is known as undercutting in the design of higher kinematic pairs, but it obviously occurs in all the application domains mentioned above. In principle, one can change either the motion, the geometry or a combination of these to eliminate the loss of contact once such a condition is detected.

However, it is intuitively clear that the development of an algorithmic approach to eliminate this condition would require knowledge not only of whether singularities exist, but also of how severe this condition is, e.g., the severity of the loss of contact. Furthermore, the knowledge of what part of the boundary of the moving object is responsible for generating these singularities during the given motion could lead to performing systematic geometric changes that eliminate this condition.

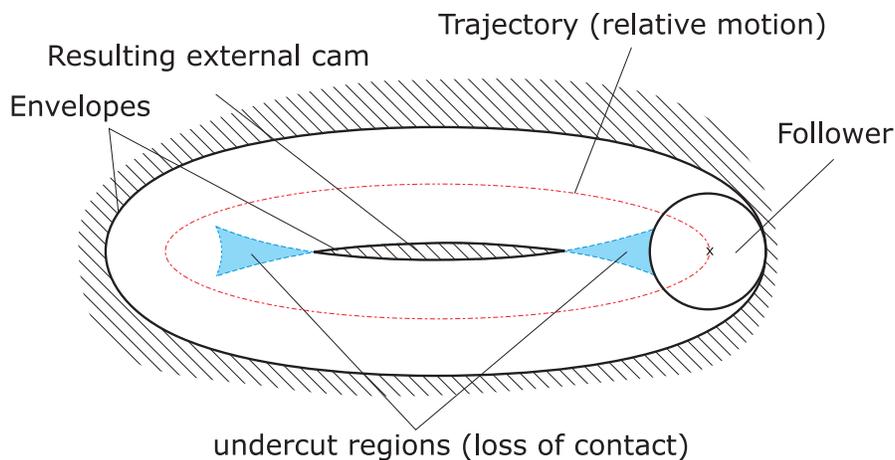


Figure 1: A cylindrical follower in relative motion generates singularities in its envelopes and therefore undercutting in the cam that is being designed. The trajectory tracks the motion of the center of the disk.

Unfortunately, the current approaches to detect these singularities are fairly specialized and

<sup>1</sup>It is worth noting that similar geometric singularities appear in problems that are characterized by moving fronts – see for example [35].

scattered across all these application domains. The more advanced techniques require the ability to compute or approximate the envelopes, and most are applicable only to restricted classes of geometries and motions, while some practical cases have no known solution (see for example [9]). The fact that these approaches provide (at most) a binary answer, even for the classes of problems that they can handle, implies that the elimination of these singularities (e.g., via changes in geometry, motion or both) is forced to rely only on the accumulated engineering experience and heuristics. On the other hand, the ability to systematically *detect*, *quantify*, and *eliminate* these singularities, and hence the corresponding malfunctions, would be extremely beneficial to all these application domains.

## 1.1 Existing Approaches to Detecting Envelope Singularities

The reasons for this *status quo* are well documented: mathematical singularities are known to create extreme difficulties, not only in the analytical realm, but also in the computational domain. These difficulties are discussed in detail in many other references (see for example [44, 2, 22, 35, 30]), so we will not focus on them here.

Broadly, there are two categories of approaches for detecting geometric singularities in the envelopes of moving surfaces. In the first category we find those (traditional) approaches that provide a closed form solution to the existence of these singularities. Unfortunately, such conditions can be derived only in those cases that place severe restrictions on the geometry and motion, which prevents them from being applicable to most “interesting” practical situations.

The second category of approaches rely on the theory on envelopes to detect these singularities, and, consequently, the associated system malfunctions. Though, these approaches are limited by the ability to compute or approximate the envelopes [25, 28, 30], which remains a difficult problem for general shapes and motions. Hence, these approaches are also forced to restrict the class of problems so that the envelopes can be approximated.

In the field of mechanical design, the loss of contact (or *undercutting*) in a higher kinematic pair has been known for over a century and represents one of the significant roadblocks in the design and manufacturing of higher pairs. Many of the earlier techniques for detecting loss of contact in the design of higher pairs fall in the first category mentioned above, and exploit the relatively simple geometry of the pairs (cylindrical or planar follower). Thus, these techniques can be applied *only* to the cases satisfying very specific and limiting geometric assumptions [8, 37]. The recent methods to design cam mechanisms have used envelope computations for cylindrical or flat faced followers [46, 47, 40]. Almost independently from these efforts, differential geometry has been used in the gear design literature to develop mathematical conditions of non-existence for envelope singularities [24, 22, 27, 41, 3, 13, 1, 7]. The derivation of these conditions relies on a theorem proposed by Litvin [23] relating the existence of the singular points to the zero valued sliding velocity at the contact point (for details see [22, 7]). All these techniques address specific classes of problems, and require difficult numerical computations, which, in turn, limit their applicability. Moreover, undercutting conditions for some classes of gearing mechanisms, such as wormgears, do not seem to be known today.

The need to detect the singularities in the envelopes occurs in other applications as well. For example, envelope singularities are used in computer aided manufacturing and computer aided process planning to describe workpiece undercutting resulting in part defects [14, 45, 32, 43, 4]. In this context, similar techniques based on the theory of envelopes have been coupled with assumptions on simple tool geometries to develop specific undercutting conditions. Singularities and their

computational properties also play a significant role in sweep boundary evaluation (see for example [25, 28, 30]); in Minkowski operations and offsets computations [34, 20]; in problems involving moving fronts [35], and in computing the geometry of shadows in computer graphics applications [10, 29, 19]. One of the recent techniques for computing offset curves and surfaces described in [33] uses rational distance maps to eliminate the portions of the envelopes that are part of the set swept by a moving object. Note that the approach discussed in [33] is aimed at computing the boundary of the sweep, and does not quantify the regions where the loss of contact occurs. On the other hand, the approach that we propose in this paper could be used not only to compute these regions of loss of contact, but also to perform sweep boundary evaluation and trimming for moving planar objects.

Observe that the recent attempts to detect the envelope singularities rely on envelope computations one way or another, and face the limitations discussed above. Importantly, none of these approaches quantifies the severity of these singularities, or of the associated malfunctions (such as the undercutting or overcutting in higher kinematic pairs), and give practically no support to the task of eliminating these singularities.

## 1.2 Outline

In this paper we focus on the class of problems that can be formulated in terms of sweeps, such as the higher kinematic pairs. We will focus here on planar *solid* shapes moving according to a non-singular affine planar motion, which is a reasonable assumption since a large number of practical situations can be abstracted to such a 2-dimensional problem (see for example the case of 2-dimensional mechanisms discussed in [18]).

Our main goals are (1) the rigorous formulation of the necessary conditions for the existence of envelope singularities applicable to arbitrarily complex planar shapes moving according to affine planar motions; (2) to show that these conditions lead not only to the detection, but also to the quantification of the malfunction induced by these singularities; (3) to demonstrate that our approach does *not* require envelope computations; and (4) to discuss how our approach can be extended to 3-dimensional problems.

We formulate our approach in section 2, and we discuss the associated computational issues in section 2.3 in the context of our prototype implementation as well as of several different examples illustrated in section 3. Finally, section 4 summarizes the contributions of this paper and explores the extension of our approach to 3-dimensional problems.

# 2 Formulation

## 2.1 Preliminaries

### 2.1.1 Motions and trajectories

Following the notation in [16], assume that motion  $M$  is a one-parameter family of transformations  $M(t)$ , and that parameter  $t$  belongs to the normalized unit interval. At every instant  $t = a$ , a set  $X$  moves to a new location in space determined by transformation  $q = M(a)$ . We will use the superscript notation to define the transformed (set of) points as  $X^q = X^{M(a)} = M(a)X$ .

The transformation  $q \in M(t)$  for some instantaneous value  $t = a$  determines the position and orientation of  $X$  relative to a fixed coordinate system. In an absolute coordinate system, every

point  $x$  of  $X$  will describe a trajectory  $T_x = \{x^q \mid q \in M\}$  in the  $d$ -dimensional Euclidean space  $\mathbb{E}^d$  in which the motion takes place. However, when observed from a moving coordinate system rigidly attached to the moving set  $X$ , the same point  $x$  will appear to describe a different trajectory denoted by  $\hat{T}_x$

$$\hat{T}_x = \{x^p \mid p \in \hat{M}\} \quad (1)$$

where  $\hat{M}$  is the *inverted motion*, i.e.,  $\hat{M}(t)$  is the inverse of  $M(t)$  for every value of  $t$  [16].

It can be shown [15] that the inverted trajectory  $\hat{T}_x$  contains **all** points  $y \in \mathbb{E}^d$  that will pass through a given point  $x$  when moved according to motion  $M$  (see one example in Figure 5):

$$\hat{T}_x = \{y \in \mathbb{E}^d \mid x \in T_y\} \quad (2)$$

### 2.1.2 Envelopes and their Geometric Interpretation

A moving object in space describes a family of shapes. By definition, the envelopes to this family of shapes are the sets of points (generally curves in 2D and surfaces in 3D) that are tangent to the family of shapes [5]. Thus, the envelopes contain *any* and *all* possible contact points with the moving object, commonly known as the generator, which implies that the boundary of *any* part that has to move in contact with the generator must be a subset of the envelopes, or otherwise contact will not occur for the prescribed motion.

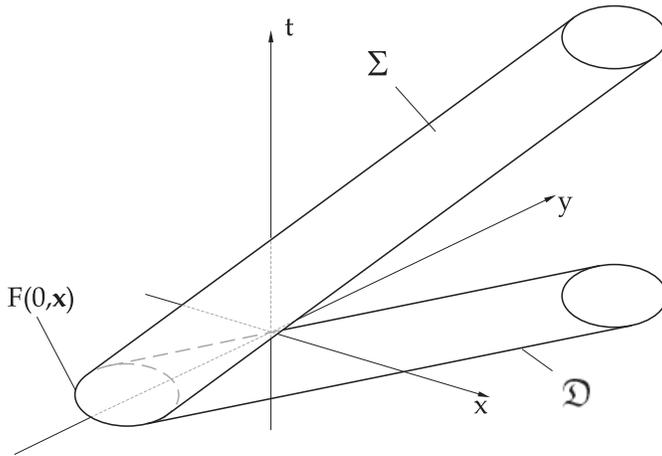


Figure 2: Envelope  $\mathfrak{D}$  is the *apparent contour* of  $\Sigma$  in the direction of  $t$ .

To illustrate the geometric significance of the envelopes, consider the family of planar disks shown in Figure 2. The envelopes to this family are well defined as the curves tangent to every disk in the family [5]. If the family is generated by a disk moving according to a planar motion, then the envelopes  $\mathfrak{D}$  are the curves tangent to the moving disk at every configuration of the motion<sup>2</sup>. From a *geometric perspective*, determining the envelope  $\mathfrak{D}$  to a shape moving in  $\mathbb{E}^d$  amounts to constructing the higher dimensional surface  $\Sigma$  in the *spacetime* domain  $\mathbb{E}^{d+1}$ , and projecting this

<sup>2</sup>We follow here the notation from [5], which also contains a good introduction to the theory of envelopes.

surface back onto  $\mathbb{E}^d$  (see also [26, 30]). Note that this spacetime domain contains the parameter  $t$  of motion  $M(t)$  as the additional dimension. By construction, the original disk is moving in this space always parallel with itself, and the envelope set  $\mathfrak{D}$  is in fact the *apparent contour* of  $\Sigma$  in the direction of  $t$  (see Figure 2 and [5]).

The set swept by a moving shape in space is defined as an infinite union operation of the object positioned at all the configurations of the motion (see for example [26, 16]). Furthermore, it is known that the boundary of the set swept (or occupied) by the moving object during a prescribed motion  $M$  is formed by a subset of the envelopes, as well as by subsets of the boundary of the object at the beginning and end configurations of  $M$  [26, 4]. Therefore, computing the boundary of the set swept by a moving object requires the computation of envelope points, which entails essentially two steps. The first one focuses on the generation of all points that are in the envelope set, which may or may not lie on the boundary of the set swept by a moving set  $X$ . The second step involves trimming those subsets of the envelope that are not part of the boundary of the sweep. Numerous boundary evaluation algorithms have been developed in solid and geometric modeling literature that implement some versions of the two steps above, such as [25, 26, 38, 4]. There are also several algorithms for computing the boundary of sweep for special cases (see for example [21]), but performing boundary evaluation for general sweeps remains difficult.

### 2.1.3 Envelope Singularities, Fold Points and Undercutting

One interesting result in the theory of singularities is due to Whitney [44], which observed that there are only two types of envelope singularities while all others are vanishing under small perturbations. The first type of singularity appears at the equatorial points of a sphere when the sphere is projected onto the plane (see also [2]). The other type of singularity, i.e., the cusp, appears when projecting surfaces such as the one shown in Figure 3 onto a plane<sup>3</sup>. In the envelope theory, the existence of cusps in  $\Sigma$  are an indication of singularities in the envelope set (i.e., the projection of  $\Sigma$  onto the original space), but detecting the existence of these cusps is a hard problem for general shapes and motions.

Except for some degenerate cases, the projection of a cusp point of  $\Sigma$  contains some special points in its neighborhood, such that more than one point of  $\Sigma$  projects (or *folds*) onto each such point. This is illustrated in Figure 3 where the cusp is labeled as  $P'_3$ , its projection onto  $\mathbb{E}^d$  is  $P_3$ , and  $P_2$  is a fold point of the plane. Observe that in this case there are three points of  $\Sigma$  that project onto the same “fold” point  $P_2$ .

Note that our *fold* and *regular* points, formally defined in [11], belong to the original plane in which the motion takes place. By contrast, the mathematical terminology of regular and fold points refers to specific points that *belong* to surface  $\Sigma$  in the  $d+1$  dimensional space [5, 44]. While this can potentially be a source of confusion, our specific choice of terminology is rooted in the fact that our fold points are obtained by projecting the apparent folds of surface  $\Sigma$  (see Figure 3) onto the plane. It is also important to observe that we only use surface  $\Sigma$  to describe our approach, and that all our computations take place in a space one dimension lower than the space where  $\Sigma$  lies.

A critical observation that we make here is that the foldings of  $\Sigma$  in the neighborhood of these cusps are responsible for generating the areas (and the envelopes) where loss of contact, or

---

<sup>3</sup>Such a surface is also called a *catastrophe surface*. The term originates in the catastrophe theory [2] and illustrates the transformation of continuous and smooth events into discrete ones. In particular, the catastrophe surface illustrates the singularities obtained by projecting a smooth surface onto a plane, which was the fundamental focus of Whitney’s landmark paper [44].

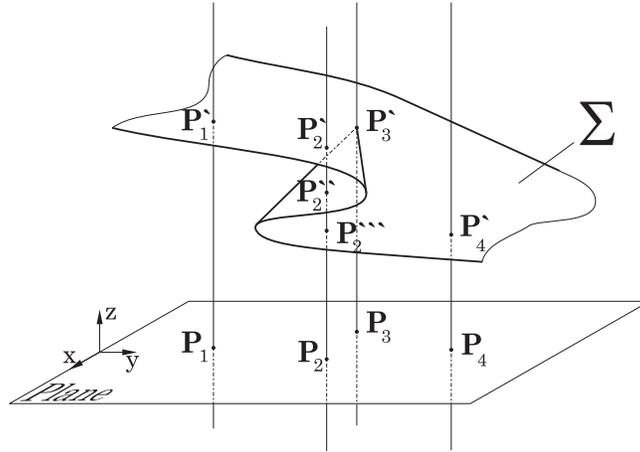


Figure 3: Projecting a surface onto a plane: Points  $P_1$  and  $P_4$  are regular points of the plane, i.e., they have only one inverse image in  $\Sigma$ ; point  $P_3$  is the projection of the cusp point of  $\Sigma$ , and point  $P_2$  is a “fold” point of the plane, which has multiple inverse images in  $\Sigma$  that “fold” onto  $P_2$ . Formal definitions of regular, and fold points in the plane are given in [11].

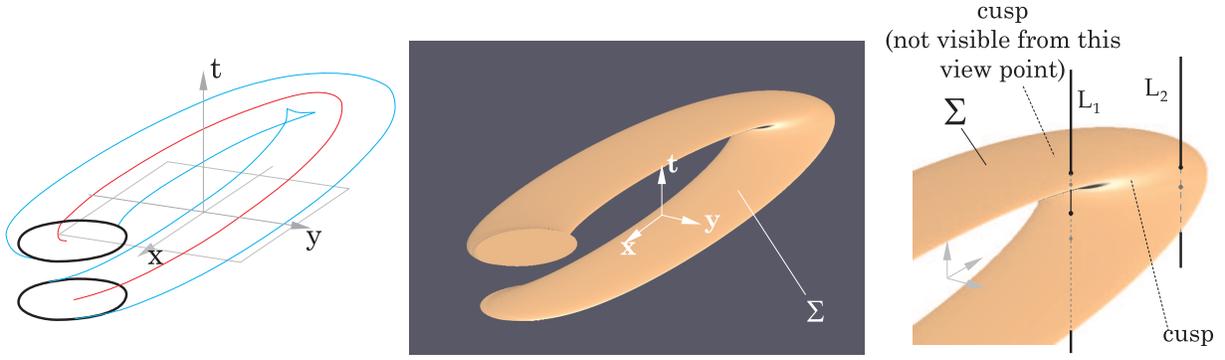
undercutting, will occur. This is clearly seen in Figure 4 which shows a disk translating<sup>4</sup> along a prescribed curve. If one constructs the  $\mathbb{E}^{d+1}$  hyperspace, where the extra dimension is the parameter  $t$  of motion  $M$ , then disk  $S$  is sweeping a well defined set  $\Sigma$  in this hyperspace as it maintains a constant orientation perpendicular to the  $t$  axis. The projection of surface  $\Sigma$  onto the  $\mathbb{E}^d$  space is the set  $\text{sweep}(S, M)$  bounded by envelope  $\mathcal{D}$ . In this example  $\Sigma$  folds onto itself near the cusp during the projection.

Our discussion suggests that the loss of contact due to envelope singularities can intuitively be defined in terms of the set of fold points in  $\mathbb{E}^d$ . For the case shown in Figure 4, if this set of fold points is empty, there is no loss of contact for the given shape and motion. Consequently, a non-empty set *may* indicate that loss of contact is present, and the size of the set quantifies the severity of this loss of contact.

Not only that for every cusp point there are infinitely many fold points, but detecting fold points appears to be conceptually and computationally more straightforward than detecting singularities. We note that for the case shown in Figure 4, one can determine these fold points by intersecting vertical lines with surface  $\Sigma$ , and by counting the number of intersections. If a given line intersects this surface more than twice, surface  $\Sigma$  will fold onto itself, thus generating the undercut area. This is illustrated in Figure 4(c) where line  $L_2$  passing through a “regular point” intersects surface  $\Sigma$  only twice, while a vertical line  $L_1$  passing through a fold point intersects  $\Sigma$  exactly four times.

In the next section we will see that this condition is necessary, but not sufficient to determine the fold points that generate loss of contact.

<sup>4</sup>We only consider translations in Figure 4 for illustrative purposes, but these “folds” in  $\Sigma$  occur for general one-parameter motions as well.



(a) The disk generates a 3D solid in the higher dimensional space  $(x,y,t)$ . (b) The 3D solid of Figure (c) shown as a shaded solid model. (c) Lines passing through fold points have four points of intersection with  $\Sigma$

Figure 4: Singularities in the envelope of a disks  $S$  that move along a specified curve indicate that undercutting is present. These singularities are difficult to detect for shapes and motions of arbitrary complexity.

## 2.2 Our Approach

The discussion above indicates that detecting fold points by intersecting lines with surface  $\Sigma$  would require the construction of a computer representation for  $\Sigma$ . If the original shape is 2-dimensional, then  $\Sigma$  is 3-dimensional and can be using a variety of techniques, some of which are supported by current CAD systems. However, it is beyond the ability of current CAD systems to handle the case when the object is 3-dimensional, and when  $\Sigma$  lives in a 4-dimensional spacetime domain. In this case, one could convert the original representation of the moving object to, say, an implicit representation, and construct  $\Sigma$  in an implicit form. As a result, the intersection between the 4D line and the 4D  $\Sigma$  would reduce to some numerical root-finding procedures. However, constructing an implicit representation of an object of arbitrary complexity, although possible in principle, is still an open problem today [36, 6].

### 2.2.1 The dual problem

Consider the 2-dimensional solid shape  $S$  shown in Figure 5 that moves according to a planar rigid body motion  $M$ . In the plane, set  $S$  will sweep a set  $\text{sweep}(S, M)$  – shaded in Figure 5(b), while in the higher dimensional space, the same set  $S$  will generate a *solid* bounded by surface  $\Sigma$ . The fact that  $\Sigma$  bounds a solid follows from the assumption that  $S$  is solid and that the parameter of the motion increases monotonically. A vertical line parallel to the  $t$  axis passing through a point  $P_0$  of the plane may or may not intersect  $\Sigma$  depending on where point  $P_0$  is located in the plane. Since the set bounded by  $\Sigma$  is a solid, if  $P_0$  is a *regular point* then the line will intersect  $\Sigma$  exactly twice<sup>5</sup>: first when the line “enters” the set bounded by  $\Sigma$ , and second when the line “exits” this set. Based on the same arguments, if  $P_0$  is a fold point, then a vertical line passing through  $P_0$  will intersect  $\Sigma$  an even number of times. If  $P_0$  is on the envelope, the number of intersections between

<sup>5</sup>See [11] for a detailed discussion of all the possible intersection cases.

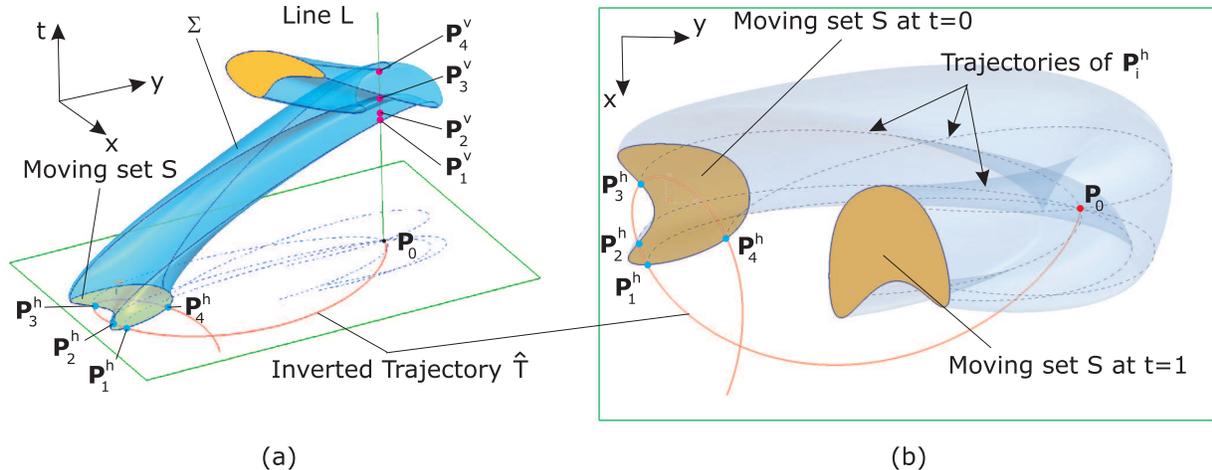


Figure 5: The duality between the *line- $\Sigma$*  intersection and the *inverted trajectory-original solid* intersection. This rigid body motion consists of both translation along a prescribed curve and rotation by  $\pi/2$ . The forward trajectories are shown as dotted curves, **all passing through point  $P_0$** .

the line and  $\Sigma$  will still be even<sup>6</sup>, but the line will also be tangent to  $\Sigma$  at one or more points.

Let's examine the geometric significance of each such intersection point  $P_i^v$  between the vertical line and  $\Sigma$ . Since each such point lies on  $\Sigma$ , which is generated by points on the boundary of  $S^0$ , then there exists a subset of the boundary of  $S^0$  whose elements pass through  $P_0$  for *some* values of  $t$ . Moreover, different values of  $t = a_i$  are associated with each intersection point, and these values represent the coordinates of each  $P_i^v$  in the direction of  $t$ . In other words, there exists only one point on the boundary of  $S^0$  that will overlap with each  $P_i$  for each of these parameter values  $a_i \in [0, 1]$ .

The above discussion implies that detecting intersection points  $P_i^v$  on  $\Sigma$  that correspond to a point  $P_0$  in the plane is *equivalent* to determining the boundary points of  $S^0$  that will pass through  $P_0$  at *some* parameter value  $t$  during the motion. The key observation is that, based on the properties of the inverted trajectories outlined in section 2.1, these boundary points of  $S^0$  must lie on the inverted trajectory passing through point  $P_0$ . These points are labeled as  $P_i^h$  in Figure 5.

Thus, this duality recasts the problem of detecting fold points in terms of straightforward *curve  $\rightarrow$  2-dimensional solid* intersections against the original boundary representation of shape  $S^0$ . As discussed in section 2.1, the trajectories of all these boundary points  $P_i^h$  of  $S^0$  will pass through point  $P_0$ , which is illustrated in Figure 5(b). Moreover, the set of all the fold points are the *fold regions* corresponding to the given shape and motion, while the boundaries of these fold regions are the *fold boundary* points.

Importantly, our approach relies on a Point Membership Classification (PMC) test for the fold regions (formally defined in [11]), which implies that *any* geometric property of these fold regions can, in principle, be computed [31].

<sup>6</sup>Note that number 0 is an even number.

### 2.2.2 The Functional Contacts

The existence of fold points, by itself, does not imply that complete loss of contact will occur between the moving object and its envelope set. In other words, the existence of fold points is a necessary, but not sufficient condition for the complete loss of contact between a moving object and its envelopes.

To see why, consider the circular disk that rotates around its “center” as shown in Figure 6. The notch will generate fold points, yet loss of contact between the disk and its envelopes does not occur. In this case, the explanation is in fact surprisingly intuitive: the notch can never contact the envelopes for this motion so the fact that the notch generates a fold region has no implications on the contact. Another example is shown in Figure 8(a) where the object maintains some contact points with its envelopes at every configuration, despite the three fold regions (see also section 3).

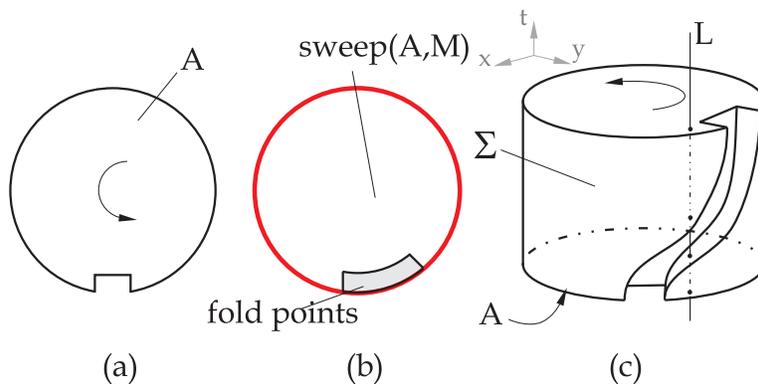


Figure 6: Fold points may exist without singularities in the envelope.

More generally, observe that contact is a local “event” and that the same object may have multiple contacts, i.e., contacts along disjoint subsets of the boundary of the object. For general motions, some of these contacting regions between a moving object and its envelope set may appear and disappear while the object maintains *some* contact points with its envelope set. However, in many engineering applications only a subset of these contacts will be functional, (i.e., will contribute to the function of the system) while the remaining contacts will be non-functional. Thus, the fact that these non-functional contacts vanish should have no consequence on the functionality of the system.

These arguments naturally lead to the question of what makes a specific contact point functional. While the answer to this question will be application dependent, in this paper we consider that a contact is functional if it is prescribed by the design engineer, so in this sense it captures the design intent. Thus we assume that a system malfunction will occur if contact is lost between the prescribed contact boundary and the corresponding envelopes to the moving object. In other words, if a fold region is “generated” by the prescribed contact boundary, then functional loss of contact occurs. Otherwise, that particular fold region has *no* influence on the functional loss of contact. While the above statement seems intuitive, stating it formally is far from trivial and requires a careful definition of what we mean by a fold region being “generated” by a prescribed subset of boundary points.

The practical implications of these observations are important. A good design engineer that designs higher pairs can identify the expected contact boundary within the pair and can therefore specify it, or at least in principle. For example a designer would specify the circular arc in Figure 6 as a contact boundary but not the notch. Thus, if loss of contact is detected for given shapes, motions and contact specifications, then one can change either the contact boundary, the motion, the geometry or a combination of these in order to eliminate the loss of contact.

By intersecting the inverted trajectories with the boundary of the moving set, we can specifically *identify* the points  $P_i^h \in \partial S$  that move through a given fold region of the space. Establishing a correlation between these points and specific loss of (or changes in) contacts – as described above – will provide the subset of  $\partial S$  that is responsible for specific changes in the contact between  $S$  and its envelopes. In turn, this can further enable systematic geometric changes to the moving shape  $S$ . Our current work focuses on developing sufficient conditions for detecting functional loss of contact that would complement the necessary conditions in terms of the existence of fold regions. However, the inverted trajectory – solid intersections induce a decomposition of the plane into [11]

- regular, fold and fold boundary points that are in the interior of the set swept by the moving object,
- sweep boundary points, and
- points that are out of the sweep.

Note that this decomposition can be used to interactively study the contact between the moving object and its envelopes as illustrated in section 3.

### 2.3 Main Algorithm and Some Computational Issues

Based on the above discussion, we can construct an algorithm to detect and quantify the fold points of a planar solid shape moving according to a non-singular affine planar motion that relies on intersection tests between the inverted trajectories and the original representation of the shape. Our algorithm also outputs the boundary points  $P_i^h$  of  $S^0$  that will pass through each detected fold region during  $M$ . The main steps of our algorithm are shown below.

Clearly, the fact that our approach relies on a PMC test for the fold points implies that these regions can be computed either exactly (within the machine precision) or approximately depending on what strategy is used to sample the points in the plane. Furthermore, given a motion  $M$ , one can generate exact or approximate trajectories as well as inverted trajectories for any point in the space. The curve-surface intersection between  $\hat{T}$  and  $\partial S$  will depend on the representations used for both  $\hat{T}$  and  $S$ . If the solid is represented by its boundary representation, then the curve-boundary intersection algorithm reduces to curve-face intersection between the inverted trajectory and the corresponding faces of the solid. This is a standard operation in CAD that is efficiently implemented in all commercial CAD systems. Furthermore, finding the number of intersections and tangencies between the inverted trajectory and the original shape representation (step 5 in algorithm 1) amounts to careful neighborhood analysis for the intersection points  $\hat{T}_{\mathbf{x}} \cap \partial S$ . This is a critical step as it must handle the degenerate cases as well as the corresponding numerical errors.

Once the neighborhood analysis is carried out and the tangencies are identified for each point  $\mathbf{x} \in \hat{T}_{\mathbf{x}} \cap \partial S$ , the classification of each  $\mathbf{x}$  from step 6 of the algorithm follows the point membership test for general sweeps outlined in [11]. The set of all fold boundary points will form the boundary

---

**Algorithm 1:** An outline of the algorithm for detecting and quantifying fold points in the plane for a moving solid object.

---

**input** : Solid planar shape  $S$ , non-singular affine planar motion  $M$   
**output:** *Non-empty fold regions* generated by  $\partial S$  throughout  $M$ ; the subsets of boundary points of  $\partial S$  that are passing through each fold region

- 1 Compute a point sampling  $\mathbf{PS}$  of the plane in which the motion takes place;
- 2 **for** every point  $\mathbf{x} \in \mathbf{PS}$  **do**
- 3     Compute the inverted trajectory  $\hat{T}_{\mathbf{x}}$  that corresponds to the inverted motion  $\hat{M}$ ;
- 4     Compute  $\hat{T}_{\mathbf{x}} \cap \partial S$ ;
- 5     Find the number of intersections  $i_{\mathbf{x}}$  and tangencies  $t_{\mathbf{x}}$  between  $\hat{T}_{\mathbf{x}}$  and  $\partial S$ ;
- 6     Classify  $\mathbf{x}$  based on the values of  $i_{\mathbf{x}}$  and  $t_{\mathbf{x}}$  as either an *interior fold point*, a *fold boundary point*, a *regular point*, a *sweep boundary point*, or a point *exterior* to the set swept by  $S$  during  $M$ ;
- 7 **end**
- 8 Mark the corresponding points  $P_i^h$  on  $\partial S$  for each detected fold region;

---

of the fold regions, which are not necessarily connected (see also section 3). Constructing the boundaries of these fold regions can follow any of the established boundary evaluation algorithms available for point sets or point clouds that will result, for example, in some tessellation (piecewise linear approximation) or an appropriate polynomial approximation. Furthermore, the inverted trajectories of those points that are in the boundary of the set swept by the moving  $S$  will have zero intersections and one or more tangencies with  $S^0$ . It is worth noting that the same sampling of the plane used to detect the fold regions can be used to generate points that are on the boundary of  $\text{sweep}(S, M)$ .

The computational cost associated with our approach depends on both the efficiency of the specific intersection test, as well as on how many times this test is actually performed. Therefore, an efficient implementation of this algorithm must not only take full advantage of the specific representation of the solid to make the intersection as efficient as possible, but must also reduce the number of times the intersection is carried out by performing adaptive sampling strategies of the plane in which the motion takes place. Standard hierarchical sampling techniques, such as quadtrees/octrees, and other spatial partitioning techniques [12], as well as other adaptive sampling methods that may use a combination of seed points in space and spatial partitioning can significantly reduce the number of points for which intersection is carried out. However, developing specific sampling techniques, as well as conditions to obtain samples that are “sufficiently dense” are outside the scope of this paper. At the same time, our approach is inherently parallel, since our tests are defined pointwise, which implies that parallel algorithms can potentially achieve impressive speedups, particularly for the cases when the adaptive sampling used is independent of steps 3-6 of the algorithm described above.

### 3 Examples

In this section we describe several examples that verify our prototype implementation and illustrate some of the main capabilities of our approach. We implemented the algorithm described above in

the Parasolid Workshop prototyping tool by UGS using Microsoft Visual Studio .NET.

The first example in Figure 7(a) shows the fold regions generated by a planar non-convex shape moving according to a planar rigid body motion, and illustrates the inverted trajectories for representative points from various fold region. In Figure 7(b) we show the computed boundary points of  $S$  that will pass through three distinct fold regions indicated in the figure. The last example illustrated in Figure 7(c) shows a deformable shape moving according to an affine (non-rigid) motion and illustrates the computed fold regions. In Figure 8(a) we show an essentially rectangular shape moving according to a motion composed of a linear translation and a rotation by  $\pi$  around its geometric center. This example identifies the subset of points of the boundary of the moving set that move through the identified fold regions. The last example in Figure 8 shows the effect of considering only the outer boundary of the triangular domain in computing the fold regions.

In all these examples we employed a simple sampling of the plane along an orthogonal grid, and used our code that implemented the steps outlined in algorithm 1 to identify the regular, fold and fold boundary points generated by  $S$  during  $M$ . As argued above, adaptive sampling strategies would significantly reduce the number of sampled points, and therefore the number of curve/boundary intersections performed. In this example implementation, the inverted trajectories were computed by sampling a dense set of points along each curve and by fitting a cubic B-spline curve interpolating the set of points.

In the example shown in Figure 7(a), the centroid of the non-convex 2D shape  $S$  moves according to an elliptical trajectory, while the shape rotates uniformly around its centroid by  $2\pi/3$ . The prescribed shape and motion generate the disjoint fold regions that are shown in the same figure. Observe that some of these fold regions are generated by the non-convex subset of the boundary of  $S$ , and that this non-convex subset *cannot* contact the envelopes for the prescribed motion. In Figure 7(a), several fold points are chosen to illustrate the intersection of their inverted trajectories with shape  $S^0$  (at  $t = 0$ ), and they all have an even number of intersections (either 4 or 6 in this example), but no tangencies, with  $\partial S$ .

For the same example, we selected three disjoint fold regions, shown in detail  $A$ ,  $B$  and  $C$  of Figure 7(b), and computed the points of the boundary  $\partial S$  of  $S$  that will pass through each of the three fold regions. Intuitively, if these points are not part of the actual boundary of contact between  $S$  and its envelopes, then these fold regions will have no influence on the functional contacts during  $M$ . Our current work focuses on understanding the relationship between these points, the corresponding fold regions and their influence on the functional contacts.

In the example shown in Figure 7(c), the centroid of disk  $S$  translates along trajectory  $T$ , while the disk rotates non-uniformly by  $2\pi/3$  around its centroid, and undergoes a non-uniform scaling. To illustrate the effect of the prescribed scaling, we superimposed a number of configurations of set  $S$  as it moves and deforms according to the affine motion. The set swept by the deformable disk  $S$  contains fold, fold boundary, as well as regular points (in the interior of  $\text{sweep}(S, M)$ ) or points that are part of the boundary of  $\text{sweep}(S, M)$ .

In the fourth example, the rectangle with rounded corners translates in the  $x$  direction along a linear trajectory  $T$ , while the rectangle rotates around its centroid by  $\pi$ . The intermediate positions were shown in order to illustrate the motion. This solid shape generates three separate fold regions during the prescribed motion marked as  $A$ ,  $B$  and  $C$ . Each of these fold regions corresponds to different subsets of the boundary of  $S$  labeled as  $\partial S_A$ ,  $\partial S_B$ , and  $\partial S_C$  that contain points passing through each one of these fold regions. Again, each of these subsets seems to be responsible for

specific changes in the local contact condition: for example  $\partial S_C$ , which is a connected curve, is clearly related to the loss of contact that happens between  $\partial S_C$  and the envelope set (when  $S$  is near the vertical configuration in this figure). On the other hand, the influence of  $\partial S_A$  and  $\partial S_B$  on the contact is far less apparent for the parameter interval used to generate the motion. In this example, the two fold regions  $A$  and  $B$  “anticipate” the loss of contact outside of  $t \in [0, 1]$ . We note that both  $\partial S_A$  and  $\partial S_B$  contain disconnected sets of boundary points, and that  $\partial S_C$  is adjacent to the boundary of  $\text{sweep}(S, M)$ , while both  $\partial S_A$ , and  $\partial S_B$  are completely enclosed in the interior of the set swept by the moving object.

In the last example, the hollow triangle with round corners  $S$  moves according to a planar rigid body motion that contains a rotation around the triangle’s centroid by  $\pi/2$ . By selecting all the faces of the triangle (including the hole) as a boundary of contact we obtain the fold regions illustrated in Figure 8(b). Eliminating the boundary of the hole from the boundary of contact changes the fold regions as indicated in the same figure.

Instead of performing the inverted trajectory – solid intersections, one could sweep the 2D solid in the 3D space time domain described in section 2.1.2, for example within a commercial CAD package. This would provide an approximation of the 3-dimensional surface  $\Sigma$ , which can then be projected onto the plane of the motion to obtain the fold regions. Note that this projection would practically introduce a second level of approximation in the result. In general, constructing  $\Sigma$  in 3D and projecting it onto the plane in *any* CAD environment would involve fairly significant approximations particularly for complex planar shapes and motions. These approximations would translate into geometric errors of the computed fold regions, which are hard to quantify mathematically since they greatly depend on the local properties of the motion and geometry, but are easily detectable with our approach. Furthermore, these computations are simply not supported by current CAD technology when  $\Sigma$  exists in a 4-dimensional space (see also section 2.2.1). On the other hand, our approach does not require envelope computations, and it therefore completely avoids all such approximations.

## 4 Conclusions

The mathematical envelopes to families of both rigid and non-rigid moving shapes play a fundamental role in a variety of problems from very diverse application domains, from engineering design and manufacturing to computer graphics and computer assisted surgery. Geometric singularities in these envelopes are known to induce malfunction or unintended system behavior, and the corresponding theoretical and computational difficulties induced by these singularities are not only massive, but also well documented.

The current techniques to detect these singularities are fairly specialized, and are applicable only to restricted classes of geometries and motions, while some practical cases have no known solution. Even when they work, these approaches provide only a binary answer, but are not capable of quantifying the severity of the corresponding malfunctions. Consequently, the elimination of these singularities (e.g., via changes in geometry, motion or both) is forced to rely only on the accumulated engineering experience and heuristics. Clearly, the ability to *detect*, *quantify*, and *eliminate* these singularities, and hence eliminate the corresponding malfunctions, would be extremely beneficial to all these application domains.

The contributions of this paper can be summarized as follows.

- We have shown that the problem of detecting and quantifying the geometric singularities in

the envelopes of moving planar objects can be reframed in terms of fold points and fold regions in the neighborhood of these geometric singularities. The underlying formalism relies on a well defined duality between the 2-dimensional Euclidean space in which the motion takes place and a  $2 + 1$  spacetime domain. We have argued in section 1 that focusing on planar problems is practical, since a large number of problems can be abstracted to the 2-dimensional domain.

- Furthermore, our approach does not require any envelope computations, and reduces to inherently parallel tests against the original geometric representation in the 2-dimensional Euclidean space in which the motion takes place. Importantly, this approach relies on a Point Membership Classification (PMC) test, which provides complete geometric information about the fold regions (see section 2.2.1). Furthermore, we have shown that our approach can be implemented within commercial geometric kernels.
- The existence of fold points, by itself, does not imply that total loss of contact will occur between the moving object and its envelope set, as discussed in section 2.2.2. However, our formalism specifically identifies the points of  $\partial S$  that move through a given fold region of the space. Establishing a correlation between these points and specific loss of (or changes in) contacts will provide the subset of  $\partial S$  that is responsible for specific changes in the contact between  $S$  and its envelopes, and can further enable systematic geometric changes to the moving shape  $S$ .
- Our decomposition of space into “regular”, “fold”, “fold boundary”, “sweep boundary” and “exterior” points for a specific object  $S$ , and motion  $M$  (see section 2.2.2) can be used for interactive explorations of contact conditions between the moving object and its envelopes. If no fold region exists for prescribed  $S$  and  $M$ , then  $S$  does not lose contact with its envelopes. On the other hand, the existence of loss of contact can be established interactively once fold regions are detected, but the automation of this process requires the formulation of sufficient conditions, as discussed in section 2.2.2.
- Once the existence of loss of contact is established, one can change either  $S$ ,  $M$ , or a combination of these, in order to eliminate the envelope singularities. Because our approach does not only detect, but it also quantifies these fold regions, it sets the stage for the development of algorithmic approaches to the *elimination* of the induced loss of contact and of the corresponding envelope singularities.

From a practical point of view, our approach provides the first steps towards a generic and automated technique to detect and quantify the loss of contact between a moving planar generator, and any object that may come in contact with the generator. In the context of mechanism design and manufacturing these malfunctions take the form of loss of contact or of violations of prescribed motions. At the same time, the same concept can be used to detect and quantify malfunctions in a number of other applications, such as manufacturing path planning, and computer assisted surgery. In geometric modeling, our approach can be used to formulate new algorithms (or augment existing ones) for evaluating the boundaries of sweeps, offsets and Minkowski operations as well as novel and efficient collision detection algorithms.

In order to preserve the generality of our approach, we avoided making restrictive assumptions on the geometries and motions that have been considered in this paper. The generality of

our approach implies that a number of specific issues have not been discussed here, and have to be addressed separately. Among these issues are set-theoretic and topological properties of the fold regions, specific representations for motions and trajectories (which in turn influence the efficiency, accuracy and robustness of the associated computations), efficient strategies for sampling the Euclidean space, as well as a number of application specific issues.

Even though we restricted here the class of problems to those that can be abstracted to the planar domain, our approach can be generalized to 3-dimensional objects moving according to general non-periodic and non-singular affine motions. Observe that a 3-dimensional moving shape will generate a 4-dimensional surface  $\Sigma$ , while the inverted trajectories will be 3-dimensional curves. In this case, detecting and quantifying the fold points and regions will reduce to 3-dimensional curve-3D solid intersections similar to those explained in this paper, and these computations can still be implemented in commercial geometric kernels. However, steps 5 and 6 of algorithm 1, and, in particular, the neighborhood computations, will have to account for 3-dimensional neighborhoods as well as degenerate cases that are specific to 3-dimensional problems.

## Acknowledgments

This work was supported in part by a grant from the University of Connecticut Research Foundation and by the National Science Foundation grant CMMI-0555937. All 3D examples were created by using Parasolid and Unigraphics, courtesy of UGS. The authors are grateful to the anonymous reviewers for their constructive suggestions.

## References

- [1] J. Argyris, F. Litvin, Q. Lian, and S.A. Lagutin. Determination of envelope to family of planar parametric curves and envelope singularities. *Computer Methods in Applied Mechanics and Engineering*, 175(1–2):175–187, 1999.
- [2] V. I. Arnol’d. *Catastrophe Theory*. Springer Verlag, Berlin / Heidelberg, Germany, third english edition edition, 2004.
- [3] B.W. Bair. Computer aided design of elliptical gears with circular-arc teeth. *Mechanism and Machine Theory*, 39(2):153–168, 2004.
- [4] D. Blackmore, M.C. Leu, and L. Wang. The sweep-envelope differential equation algorithm and its application to nc machining verification. *Computer Aided Design*, 29:629–637, 1997.
- [5] James W. Bruce and Peter J. Giblin. *Curves and Singularities*. Cambridge University Press, 1992.
- [6] Suzanne F. Buchele and Richard H. Crawford. Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. In *Proceedings of the 8<sup>th</sup> ACM Symposium on Solid Modeling and Applications*, pages 135–144. ACM Press, 2003.
- [7] S.L. Chang, C.B. Tsay, and L.I. Wu. Mathematical model and undercutting analysis of elliptical gears generated by rack cutters. *Mechanism and Machine Theory*, 31(7):879–890, 1996.

- [8] Fan Y. Chen. *Mechanics and Design of Cam Mechanisms*. Pergamon Press, 1982.
- [9] J R. Colbourne. Undercutting in worms and worm gears. *AGMA - American Gear Manufacturers Association, Technical Paper*, 93FTM1, 1993.
- [10] L. Donati and N. Stolfi. Singularities of illuminated surfaces. *International Journal of Computer Vision*, 23(9):207–216, 1997.
- [11] H. Erdim and H.T. Ilieş. A point membership classification for general sweeps. In *ASME Design Automation Conference, IDETC2007*, Las Vegas, NV., September 4-7 2007.
- [12] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition.
- [13] Z.H. Fong, T.W. Chiang, and C.W. Tsay. Mathematical model for parametric tooth profile of spur gear using line of action. *Mathematical and Computer Modelling*, 36(4-5):603–614, 2002.
- [14] P.J. Gray, F. Ismail, and S. Bedi. Graphics-assisted rolling ball method for 5-axis surface machining. *Computer-Aided Design*, 36:653–663, 2004.
- [15] H.T. Ilieş. Continuous collision and interference detection for 3D geometric models. Technical report, Computational Design Laboratory, University of Connecticut, December 2006.
- [16] H.T. Ilieş and V. Shapiro. The dual of sweep. *Computer Aided Design*, 31(3):185–201, March 1999.
- [17] H.T. Ilieş and V. Shapiro. A class of forms from function: the case of parts moving in contact. In *Proceedings of the ASME 2001 DETC, 13<sup>th</sup> International Design Theory and Methodology Conference*, 2001.
- [18] L. Joskowicz and E. Sacks. Computer-aided mechanical design using configuration spaces. *IEEE, Computers in Science and Engineering*, 1(6):14–21, 1999.
- [19] D.C. Knill, P. Mamassian, and D. Kersten. Geometry of shadows. *J. Opt. Soc. Am. A*, 14(12):3216–3232, 1997.
- [20] I.K. Lee, M.S. Kim, and G. Elber. Polynomial/rational approximation of minkowski sum boundary curves. *Graph. Models Image Process.*, 60(2):136–165, 1998.
- [21] J.H. Lee, S.J. Hong, and M.S. Kim. Polygonal boundary approximation for a 2d general sweep based on envelope and boolean operations. *The Visual Computer*, 16(3-4):208–240, 2000.
- [22] F. Litvin. Development of gear technology and theory of gearing. Technical Report NASA RP1406/ARLTR1500, NASA, 1997.
- [23] Faydor L. Litvin. *Theory of Gearing*. Nauka, 2<sup>nd</sup> edition, 1968. in Russian.
- [24] Feyodor L. Litvin. *Gear Geometry and Applied Theory*. Prentice-Hall, 1994.
- [25] K.A. Malek, D. Blackmore, and K. Joy. Swept volumes: Foundations, perspectives, and applications. *International Journal of Shape Modeling*, 12(1):87–127, 2006.

- [26] R.R. Martin and P.C. Stephenson. Sweeping of three dimensional objects. *Computer Aided Design*, 22(4):223–233, 1990.
- [27] G.C. Mimmi and P.E. Pennacchi. Non-undercutting conditions in internal gears. *Mechanism and Machine Theory*, 35(4):477–490, 2000.
- [28] Martin Peternell, Helmut Pottmann, Tibor Steiner, and Hongkai Zhao. Swept volumes. *Computer-Aided Design Appl.*, 2:599–608, 2005.
- [29] S. Plantinga and G. Vegter. Contour generators of evolving implicit surfaces. In *ACM SM03*, pages 23–32, Seattle, Washington, USA., June 1620 2003.
- [30] H. Pottmann and M. Peternell. Envelopes – computational theory and applications. In *Spring Conference on Computer Graphics and its Applications*, pages 3–23, Budmerice, Slovakia, 2000.
- [31] A.A.G. Requicha. Representations for rigid solids: Theory, methods and systems. *Computing Surveys*, 12(4):437–463, 1980.
- [32] D. Roth, S. Bedi, and F. Ismail. Generation of swept volumes of toroidal endmills in five-axis motion using space curves. In *Symposium on 5<sup>th</sup> Solid Modeling and Applications*, Ann Arbor, MI, 1999.
- [33] J.K. Seong, G. Elber, and M.S. Kim. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *Computer Aided Design*, 38(3):183–193, March 2006.
- [34] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982. Volume 1.
- [35] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [36] V. Shapiro and D.L. Vossler. Separation for boundary to csg conversion. *ACM Trans. Graph.*, 12(1):35–55, 1993.
- [37] Joseph E. Shigley and John J. Jr. Uicker. *Theory of Machines and Mechanisms*. McGraw-Hill, 1995. 2nd ed.
- [38] A. Sourin and A. Pasko. Function representation for sweeping by a moving solid. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):11–18, March 1996.
- [39] R.H. Taylor, L. Joskowicz, and B et al Williamson. Computer-integrated revision total hip replacement surgery: Concept and preliminary results. *Medical Image Analysis*, 1(1):1–19, 1999.
- [40] D.M. Tsay and H.M. Wei. A general approach to the determination of planar and spatial cam profiles. *Journal of Mechanical Design*, 118:259–265, June 1996.
- [41] R.T. Tseng and C.B. Tsay. Mathematical model and undercutting of cylindrical gears with curvilinear shaped teeth. *Mechanism and Machine Theory*, 36(11–12):1189–1202, 2001.

- [42] J. Wallner and H. Pottmann. On the geometry of sculptured surface machining. In P. J. Laurent, C. Rabut, and L. L. Schumaker, editors, *Curves and Surfaces, Nashville, TN, 2000*. Vanderbilt Univ. Press, 2000.
- [43] K. Weinert, S. Du, P. Damm, and M. Stautner. Swept volume generation for the simulation of machining processes. *Machine Tools & Manufacture*, 44:617–628, 2003.
- [44] H. Whitney. On singularities of mappings of Euclidean spaces. I. Mappings of the plane onto a plane. *Ann. Math II, Ser 62*, pages 374–410, 1955.
- [45] X.J. Xu, C. Bradley, Y.F. Zhang, L.T. Loh, and Y.S. Wong. Tool-path generation for five-axis machining of free-form surfaces based on accessibility analysis. *International-Journal-of-Production-Research*, 40(14):3253–3274, 2002.
- [46] H.S. Yan and W.T. Cheng. Curvature analysis of roller-follower cam mechanisms. *Mathematical and Computer Modelling*, 29:69–87, 1999.
- [47] Shyue-Cheng Yang. Determination of spherical cam profiles by envelope theory. *Journal of Materials Processing Technology*, 116(2–3):128–136, 2001.

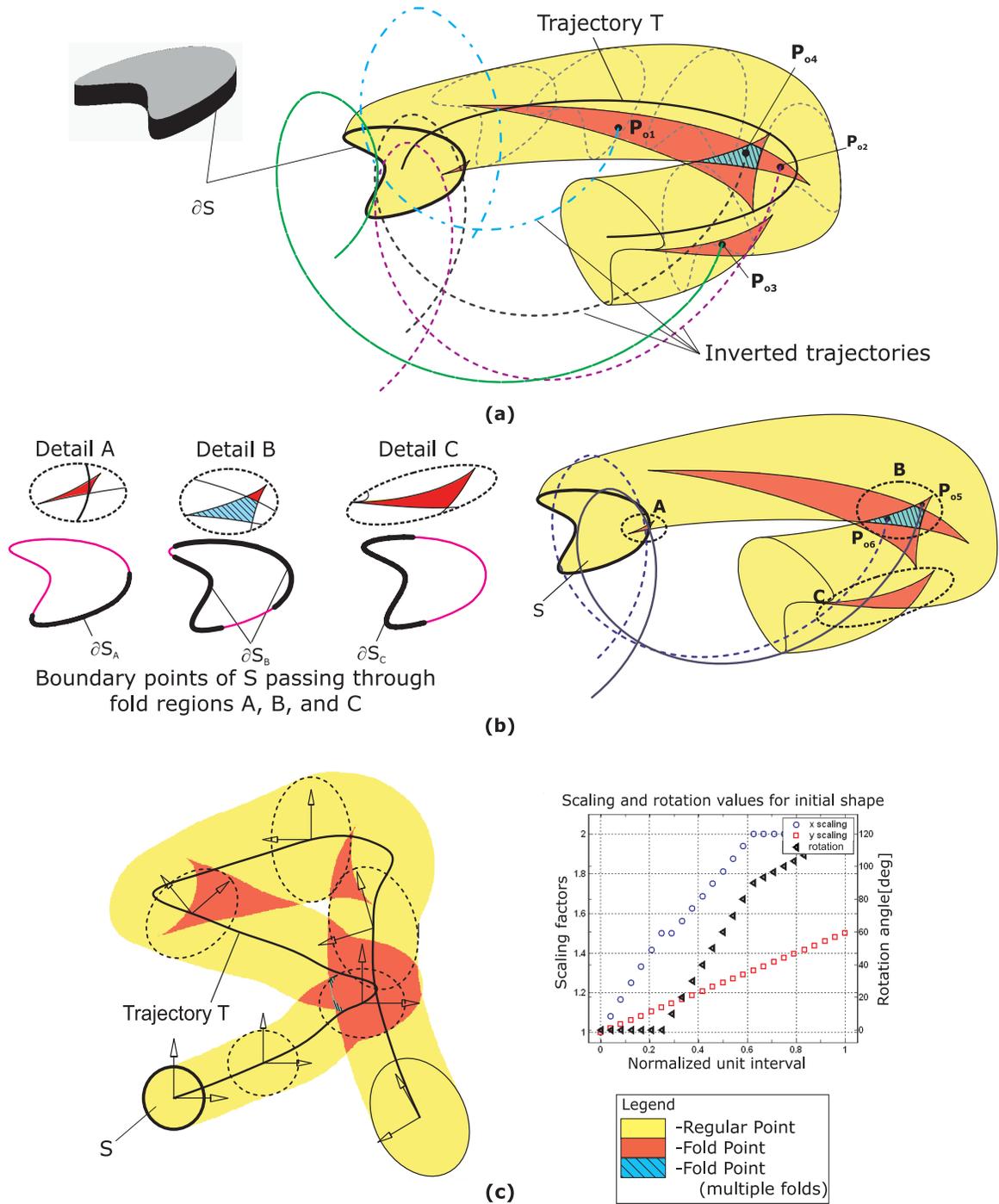
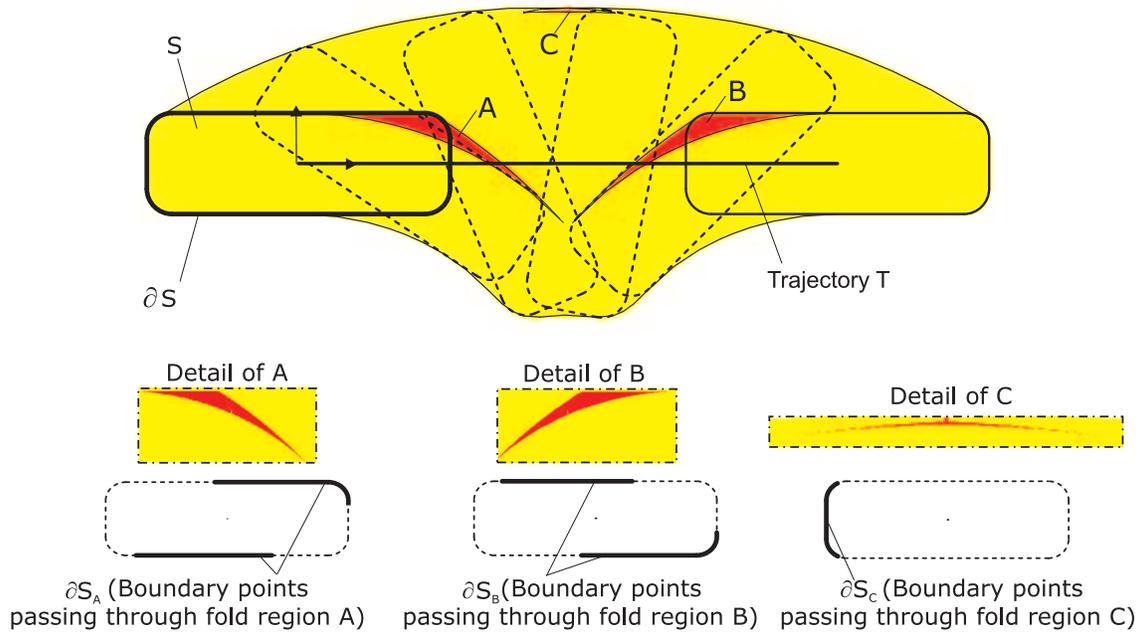
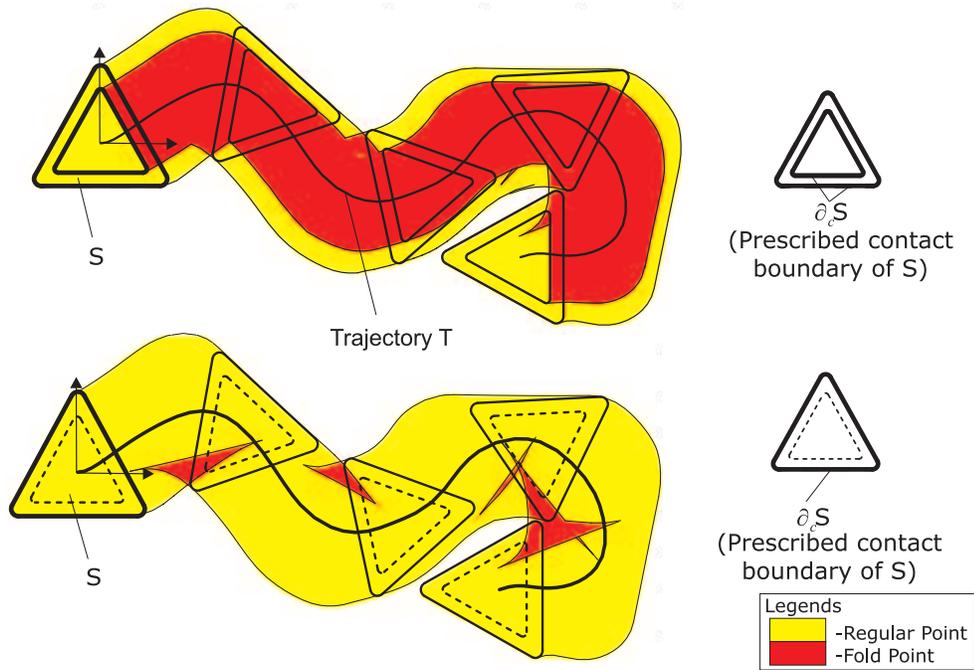


Figure 7: (a) The fold regions corresponding to a planar non-convex shape  $S$  moving according to a planar rigid body motion and selected inverted trajectories; (b) The points of  $\partial S$  that pass through specified fold regions; (c) a shape moving according to an affine motion involving translation, rotation and scaling, and the computed fold regions.



(a)



(b)

Figure 8: (a) The distinct fold regions for a planar convex shape  $S$  are generated by different subsets of the boundary of contact; (b) Reducing the boundary of contact of  $S$  changes the corresponding fold regions for a hollow triangle with round corners.