Computer-Aided Design 46 (2014) 227–232

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Technical note

Self-overlapping curves: Analysis and applications

Uddipan Mukherjee

University of California, Irvine, USA

ARTICLE INFO

Keywords:
Self-overlapping curve
Immersion

ABSTRACT

When a disk in 2D is stretched arbitrarily with possible self-overlaps, without twisting it, its boundary forms a complex curve known as a self-overlapping curve. The mapping between the disk and its deformed self, also called an immersion of the disk, is useful in many applications like shape morphing and curve interpretation. Given a self-overlapping curve, an algorithm for computing its immersion is presented, which has an average time complexity quadratic in the number of points on the curve.

1. Introduction

Consider a circular disk in 2D, painted blue on one side and red on the other. The disk is placed in a plane such that only the blue side is visible. If the disk is now arbitrarily stretched and given a multiple number of self-overlaps, without twisting it at any time, still only the blue side is visible. The boundary of this disk is called a *self-overlapping curve* [1] (Fig. 1). Trivially, every simple closed-loop curve without self-intersections is a self-overlapping curve. The homeomorphism or one-to-one mapping between the disk and its deformed self is known as an *immersion* of the disk.

Self-overlapping curves occur in many physical systems. In graphite, which is one of the crystalline forms of carbon, hexagonal rings of connected carbon atoms form a self-overlapping polygon [2] (Fig. 2). A method for computing the disk immersion gives a way of interpreting the polygon from the boundary curve laid out by the carbon atoms.

In keyframe animations, objects between keyframes are often self-overlapping polygons. Morphing between the objects by considering their boundaries only does not produce intuitive morphs [3]. Computing immersions of the source and target shapes, and morphing one immersion to another produces aesthetically better and intuitive morphs (Fig. 3). Self-overlapping curves also appear as boundaries of lanes in a freeway. Computing an immersion in such cases clearly brings out the layout of the road pattern (Fig. 4).

An algorithm for computing the immersions of a self-overlapping curve is presented in this paper. The curve is segmented into a set of simple curves (Jordan curves) by non-trivial line segments (Fig. 5). The input to the algorithm is a sequence of points, in order, forming the curve. It has an average time complexity quadratic in the number of points on the curve.

Interestingly, there can be multiple immersions of a disk corresponding to a self-overlapping curve, i.e. a disk may be overlapped in multiple ways to produce the samecurve as the boundary (Fig. 6). The immersion algorithm presented here computes all possible immersions of a given self-overlapping curve.

2. Related work

Self-overlapping curves were studied in detail by Shor et al. [1], wherein an algorithm to detect whether a closed-loop self-intersecting curve is self-overlapping was proposed. This algorithm runs in $O(n^3)$ time, where n is the number of points on the curve. For curves in general positions (also known as self-overlapping polygons), where the intersection points do not coincide with any of the curve points, this algorithm also produces an arbitrary triangulation of the curve interior. For curves with multiple immersions, [1] proposes an algorithm to compute the number of immersions in $O(n^3 \log n)$ time. Applications like shape morphing require a 'good' Delaunay-like triangulation of self-overlapping polygons to produce aesthetically intuitive morphs, which cannot be obtained immediately from [1]. A method for obtaining such a triangulation from an arbitrary one, by a series of edge flips, for a self-overlapping polygon was proposed in [3]. However, the number of edge flips required for the same is unknown. Obtaining a suitable triangulation of the curve interior catering to the needs of the application concerned is thus challenging – even more so – because no known method of triangulating a simple polygon can be easily extended to self-overlapping polygons.

Mukherjee et al. [4] present an abstract representation of a self-overlapping curve which may be helpful in computing its immersions, but does not explicitly produce one. Guo et al. [2] investigate the different immersions produced by a self-overlapping curve generated by overlapping graphite atoms. Brinkmann et al. [5] and Graver [6] generalize the problem for (m, k) patches, $(m, k \geq 3)$ where regular m-gon tiles are attached to form the patches with k tiles meeting at an interior vertex and $k-1$ tiles meeting at boundary vertices.

Given a general self-intersecting curve, Titus [7] and Blank [8] examine the condition when it is self-overlapping. However, the exact time complexities of these methods are not mentioned.

E-mail address: umukherj@ics.uci.edu.

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.08.037

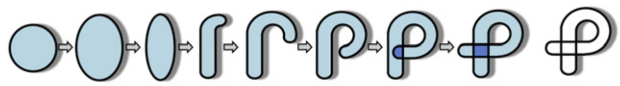

Fig. 1. A disk painted blue on the front and red on the back side is stretched and overlapped (left to right) without twisting such that only the blue side is always visible. The disk boundary is a self-overlapping curve (extreme right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The algorithm presented in this paper explicitly computes all the immersions from a given self-overlapping curve. It has an average time complexity of $O(n^2)$, where n is the number of points on the curve. The given self-overlapping curve is divided into a set of simple curves, each of which can be triangulated in any known way of triangulating simple polygons, thereby making the method suitable for applications like shape morphing.

Although not directly related to this paper, the work of Eppstein et al. [9] requires special mention where it was proved that extruding an immersion of a self-overlapping curve to a non self-intersecting surface in 3D (embedding) is NP complete.

3. Definitions

Self-overlapping curve: A self-intersecting closed-loop curve which can be divided into a set of simple curves by non-trivial line segments (Fig. 7). Traversing the curve in a particular direction (e.g. clockwise) from a starting point defines a natural interior of the curve. Every simple polygon is trivially a self-overlapping curve.

An edge of a curve is defined as the line segment joining two adjacent points on the curve. If the curve intersects in general positions only, i.e. the edges of the curves intersect only at non-terminal points, it is called a self-overlapping polygon. The formal definition of self-overlapping curves can be compared with an intuitive one introduced at the beginning of the paper.

Immersion: A continuous function $i : M \rightarrow T$ such that for any point $v \in M$ there exists a neighborhood $u(v)$ within which i restricts to a homomorphism from $u(v)$ to $i(u(v))$ [9].

A self-overlapping curve forms the boundary of a deformed disk. An immersion of a self-overlapping curve is a one-to-one mapping between the interior of the curve and the disk.

Crest point: A local extremum point with respect to the horizontal such that the curve takes a left turn at the crest point, assuming that the interior of the curve is to its right (Fig. 8). If the crest point is at a local maxima, it is called a *maximal crest point*, and if it is at a local minima, it is called a *minimal crest point*.

4. Computing immersions

Given a self-overlapping curve, horizontal rays directed towards the right (positive X axis) are drawn for each crest point P. If P has an ordinate value y, the ordinate value of the corresponding ray is $y - \epsilon$, if P is a maximal crest point, and $y + \epsilon$ if P is a minimal crest point, where ϵ is very small. These rays intersect the curve at various points including a pair of points in the local neighborhood

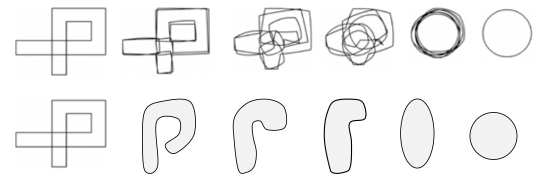

Fig. 3. Morphing boundaries only (top) and interior (bottom) of a self-overlapping polygon.

Fig. 4. An aerial view of a nested freeway. The boundaries of the lanes are self-overlapping curves (partly highlighted in red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of each crest point. The ray–curve intersection points are defined as *cuts*. If the curve crosses a ray r, the cut is termed as positive and marked as r, or termed as negative and marked as r', according as the curve crosses the ray from right to left or left to right. The ray and its corresponding cuts are marked with the same symbol but it will be clear from the context which one is referred. The cuts in the local neighborhood of a crest point (having ordinate values $y \pm \epsilon$, where y is the ordinate value of the crest point) are called crest cuts. They are referred to as maximal or minimal crest cuts according as the corresponding crest point is a maximal or minimal crest point. Traveling along the curve, two consecutive cuts i and j are called adjacent cuts (Fig. 9).

A self-overlapping curve can be potentially segmented into two self-overlapping curves, by cutting it along a pair of positive and negative cuts from the same ray. If such a pair exists, it is called a *valid* pair of cuts. However, if by cutting along a pair of positive and negative cuts yields two curves, either of which is non self-overlapping, the cut pair is invalid (Fig. 10). A cut pair is defined to be left valid if the portion of the original curve to the left of the corresponding ray is self-overlapping, and right valid if the curve to the right of the ray is self-overlapping. Thus, a valid cut pair is both left valid and right valid.

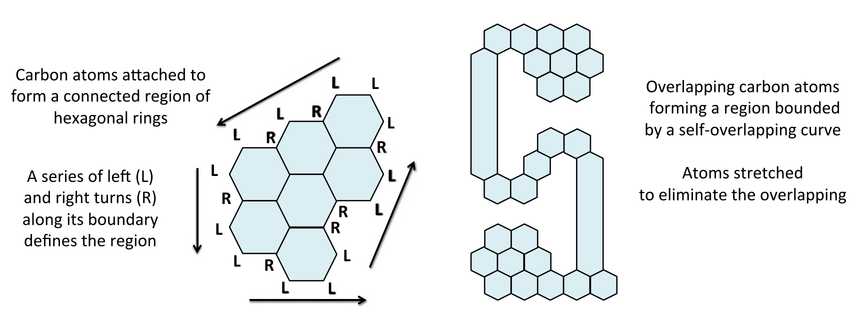

Fig. 2. Self-overlapping polygon in graphite atom.