Snap Rounding of Bézier Curves*

Arno Eigenwillig
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
arno@mpi-inf.mpg.de

Lutz Kettner†
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
kettner@mpi-inf.mpg.de

Nicola Wolpert
Hochschule für Technik
70174 Stuttgart, Germany
nicola.wolpert@hft-stuttgart.de

ABSTRACT

We present an extension of snap rounding from straight-line segments (see Guibas and Marimont, 1998) to Bézier curves of arbitrary degree, and thus the first method for geometric rounding of curvilinear arrangements. Our algorithm takes a set of intersecting Bézier curves and directly computes a geometric rounding of their true arrangement, without the need of representing the true arrangement exactly. The algorithm's output is a deformation of the true arrangement that has all Bézier control points at integer points and comes with the same geometric guarantees as in straight-line snap rounding: during rounding, objects do not move further than the radius of a pixel, and features of the arrangement may collapse but do not invert.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*geometric algorithms*; D.m [**Software**]: Miscellaneous—*robust geometric computation*

General Terms

Algorithms, Reliability, Theory

Keywords

Bézier curves, arrangement, snap rounding, geometric rounding, intersection computation, robustness, splines

1. INTRODUCTION

Putting geometric algorithms into practice is a challenging task for several well-known reasons, most prominently the delicate dependency between numerical and combinatorial computations. Often, algorithms are designed for exact arithmetic. Naive use of finite-precision arithmetic in its place is a recipe for disaster.

One solution is to design specific algorithms that can cope with numerical imprecision in a controlled manner. We mention two examples for arrangements of curves. Milenkovic and Sacks [17] compute an approximate arrangement and prove, under certain empirically justified assumptions on the underlying numerical solver, that there exists a perturbation of the input which realizes the computed arrangement. Halperin and Shelton [11] show how to actually carry out a *controlled perturbation* of the input such that fixed-precision arithmetic suffices; see also [9] and [16]. The guarantee offered by these methods bounds *backward error*: the result is correct for a slight (implicit or explicit) perturbation of the original input.

A different approach is the *Exact Geometric Computation* (EGC) paradigm coined by Yap (see [23]) that demands exact determination of geometric relations, using as much numerical precision as necessary. General-purpose geometry libraries such as CGAL [3] [14] and LEDA [15] [14] have successfully implemented EGC. Yap [24] gave an EGC algorithm for intersecting Bézier curves, solving the difficult case of tangential intersections with subdivision up to a separation bound, which may be costly. EGC delivers error-free results, but these results live on an island: Exact coordinates of newly computed objects cannot be used "as is" in traditional file formats or APIs with fixed-precision number types; it is necessary to round them. Also, EGC implementations with exact construction of objects suffer from exponential coordinate growth in cascaded constructions (see [18]) if no intermediate rounding takes place.

Our contribution follows a third approach, which we introduce by way of its classical example: Consider a planar arrangement of straight-line segments represented by a graph. Vertices are labelled by their cartesian coordinates. Edges represent the line segment between their endpoints; they may not intersect in their interiors. We want to round all vertex coordinates to, say, integers. Rounding means moving each vertex, and thus implicitly also its incident line segments. Doing that may introduce spurious crossings of segments, so a mere numerical rounding of vertex coordinates would make the geometry implied by the vertex coordinates inconsistent with the graph data. Also, it would invert the true topology. What one desires instead is a *geometric rounding* that modifies the graph data as well, such that (i) the data structure remains consistent, and (ii) the

*This work was partially supported by the IST Programme of the European Union as a Shared-cost RTD (FET Open) Project under Contract No. IST-006413 (ACS – Algorithms for Complex Shapes).

†Now at mental images GmbH, 10623 Berlin, Germany.

original topology is preserved to some extent. A method to perform such geometric rounding can be used to reduce numeric precision in an existing arrangement; e.g., to export an exact arrangement from the EGC island. Moreover, such a method gives rise to algorithms that compute a rounded arrangement directly and thus can avoid the costly exact computation of intersection points. In either case, one obtains a result with bounded *forward error*: the output is close to the true result for the given input.

The problem of rounding planar arrangements of straight-line segments, as well as its three-dimensional extensions, have been in the focus of the research literature on geometric rounding. Formulations that prescribe full preservation of topology are often NP-hard, as shown by the fundamental result on the hardness of preserving the nesting relationship of planar polygons within a given tolerance [19]. Efficient methods are known if the geometric rounding is allowed to collapse small features and to create new contacts in narrow settings. For planar arrengements, we mention the early work by Greene and Yao [7], followed by the popular *snap rounding*, due to Hobby [13] and independently Greene [6], and *shortest-path rounding* by Milenkovic [18], which can result in less additional vertices than snap rounding. Efficient implementations of snap rounding have been treated by Guibas and Marimont [8], Goodrich at al. [5], de Berg et al. [1] and Hershberger [12]. Variations of snap rounding are iterated snap rounding by Halperin and Packer [10], see also [20], and the extension to boolean operations on polygons by Devillers and Guigue [2].

Our result We extend snap rounding from straight-line segments to Bézier curves of arbitrary degree, and thus attain the first method for geometric rounding of curvilinear arrangements. We have chosen snap rounding, because it allows proof techniques that generalize well from the straight-line case to our setting. We study Bézier curves, because they are ubiquitous in applications and can represent all polynomially parameterized curves, e.g., pieces of splines. Also, it is clear how to round one Bézier curve, namely by rounding its defining control points, and how this affects the curve. (In contrast, it would not be so clear how to round a segment of an algebraic curve.) The convex hull of the control points encloses the curve, and this straight-line enclosure gives us a point of attack for the generalization from line segments.

In Section 2, we review basics on Bézier curves necessary to describe the actual algorithm in Section 3. The algorithm receives a set of input curves and subdivides them until it has approximated intersection points sufficiently to determine their rounding, and until the enclosing polygons are tight enough so that rounding succeeds. Here, subdivision serves a double purpose: locating intersection points (cf. [24] [21, §3.7]) and breaking curves, like ursegments are broken at hot pixels in straight-line snap rounding. Section 4 presents a geometric analysis of the output. We extend the arguments of [8] for straight-line snap rounding to the polygonal enclosures of the rounded Bézier curves, and thus to the rounded curves themselves. We arrive at the same geometric guarantees as known from straight-line snap rounding: During rounding, *objects do not move further than the radius of a pixel* (Theorem 8). Features of the arrangement may collapse but do not invert; in particular, *the cyclic order of non-collapsed edges around a vertex is preserved* (Theorem 18).

2. BASICS ON BÉZIER CURVES

A *Bézier curve* [21] [4] of formal degree $n > 0$ is a parameterized curve $\mathbf{b}\colon [0,1] \to \mathbb{R}^2$ written as $\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ using arbitrary *control points* $\mathbf{b}_0, \ldots, \mathbf{b}_n \in \mathbb{R}^2$ and the *Bernstein polynomials* $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$ for $i = 0, \ldots, n$. Notice that $\mathbf{b}(0) = \mathbf{b}_0$ and $\mathbf{b}(1) = \mathbf{b}_n$. The polyline $\overline{\mathbf{b}_0 \cdots \mathbf{b}_n} := \overline{\mathbf{b}_0\mathbf{b}_1} \cup \overline{\mathbf{b}_1\mathbf{b}_2} \cup \cdots \cup \overline{\mathbf{b}_{n-1}\mathbf{b}_n}$ is called the *control polygon*[1] of $\mathbf{b}(t)$. As the Bernstein polynomials on $[0,1]$ form a non-negative partition of unity, the trace $\mathbf{b}([0,1])$ of $\mathbf{b}(t)$ is a subset of the convex hull $\operatorname{conv}\{\mathbf{b}_i\}_i = \operatorname{conv}\{\mathbf{b}_0, \ldots, \mathbf{b}_n\}$ of the control points. We call $\operatorname{conv}\{\mathbf{b}_i\}_i$ the *enclosing polygon* of $\mathbf{b}$. Recall that a parameterized curve is said to be *regular* at t if the *tangent vector* $\mathbf{b}'(t)$ is non-zero.

Given $\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i^0 B_i^n(t)$ and $\alpha \in (0,1)$, one can *subdivide* $\mathbf{b}(t)$ into $\mathbf{b}^-(t) := \sum_{i=0}^n \mathbf{b}_0^i B_i^n(t) := \mathbf{b}(\alpha t)$ and $\mathbf{b}^+(t) := \sum_{i=0}^n \mathbf{b}_i^{n-i} B_i^n(t) := \mathbf{b}((1-\alpha)t + \alpha)$ with the *de Casteljau algorithm* by setting $\mathbf{b}_i^j := (1-\alpha)\mathbf{b}_i^{j-1} + \alpha \mathbf{b}_{i+1}^{j-1}$ for all $j = 1, \ldots, n$ and $i = 0, \ldots, n-j$. Unless indicated otherwise, we subdivide at $\alpha = \frac{1}{2}$. The new control polygons $\overline{\mathbf{b}_0^0 \mathbf{b}_0^1 \cdots \mathbf{b}_0^n}$ and $\overline{\mathbf{b}_0^n \mathbf{b}_1^{n-1} \cdots \mathbf{b}_n^0}$ are contained in $\operatorname{conv}\{\mathbf{b}_i\}_i$. Their concatenation $\overline{\mathbf{b}_0^0 \cdots \mathbf{b}_0^n \cdots \mathbf{b}_n^0}$ is a better polyline approximation of $\mathbf{b}(t)$ than the original control polygon. Repeating subdivision recursively on the subcurves produces a quadratically convergent sequence of polyline and polygon approximations to the Bézier curve formed by the concatenated control polygons or enclosing polygons, respectively. We look at a Bézier curve in exactly this way: it is a curve whose position is known through a *polygonal enclosure* that can be refined further if needed.

The derivative of $\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ is again a Bézier curve: As $\frac{\mathrm{d}}{\mathrm{d}t} B_i^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$, one has $\mathbf{b}'(t) = n \sum_{i=0}^{n-1} \Delta\mathbf{b}_i B_i^{n-1}(t)$ with $\Delta\mathbf{b}_i := \mathbf{b}_{i+1} - \mathbf{b}_i$. Let $\mathbf{m}$ be a unit-length vector. We call a sequence or function *increasing in direction* $\mathbf{m}$, or $\mathbf{m}$-*increasing*, if its image $\langle \mathbf{m}, \cdot \rangle$ under projection onto $\mathbf{m}$ is increasing (in the strict sense, equality is not permitted). We also say *monotone* instead of $\mathbf{m}$-increasing if any increasing direction $\mathbf{m}$ exists. We say that $\mathbf{b}(t)$ is *bcp-increasing in direction* $\mathbf{m}$, or *bcp-monotone*, if its sequence of Bézier control points $\mathbf{b}_0, \ldots, \mathbf{b}_n$ is increasing in direction $\mathbf{m}$.

LEMMA 1. *A Bézier curve* $\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ *is bcp-monotone if and only if* $0 \notin \operatorname{conv}\{\Delta\mathbf{b}_0, \ldots, \Delta\mathbf{b}_{n-1}\}$.

PROOF. If $\langle \mathbf{m}, \mathbf{b}_{i+1} \rangle > \langle \mathbf{m}, \mathbf{b}_i \rangle$, then $\langle \mathbf{m}, \Delta\mathbf{b}_i \rangle > 0$ for all $i < n$, so any convex combination of the $\Delta\mathbf{b}_i$ lies in the half-plane $\langle \mathbf{m}, \cdot \rangle > 0$; this implies $0 \notin \operatorname{conv}\{\Delta\mathbf{b}_i\}_i$. Conversely, $0 \notin \operatorname{conv}\{\Delta\mathbf{b}_i\}_i$ implies the existence of a half-plane $\langle \mathbf{m}, \cdot \rangle > 0$ comprising $\operatorname{conv}\{\Delta\mathbf{b}_i\}_i$, giving us a bcp-increasing direction $\mathbf{m}$. □

Rounding may create repeated control points $\mathbf{b}_i = \mathbf{b}_{i+1}$. Let $\operatorname{uniq}((\mathbf{b}_0, \ldots, \mathbf{b}_n))$ denote the maximal subsequence of $(\mathbf{b}_0, \ldots, \mathbf{b}_n)$ in which no two successive points are equal. We call a non-constant Bézier curve $\mathbf{b}(t)$ *weakly bcp-monotone* or *weakly bcp-increasing in direction* $\mathbf{m}$, if $\operatorname{uniq}((\mathbf{b}_0, \ldots, \mathbf{b}_n))$ is $\mathbf{m}$-increasing.

LEMMA 2. *A non-constant Bézier curve* $\sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ *is weakly bcp-monotone if and only if* $0 \notin \operatorname{conv}(\{\Delta\mathbf{b}_i\}_i \setminus \{0\})$.

[1] This is the established terminology, even though the control polygon is not a polygon but a polyline.

LEMMA 3. *Let* $\mathbf{b}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(t)$ *be a Bézier curve that is weakly bcp-increasing in direction* $\mathbf{m}$.

(i) The function $t \mapsto \mathbf{b}(t)$ *is* $\mathbf{m}$*-increasing and thus injective for* $t \in [0,1]$.
It is regular for $t \in (0,1)$*; if* $\mathbf{b}(t)$ *is bcp-increasing, it is regular for* $t \in [0,1]$.

(ii) The endpoints $\mathbf{b}(0)$ *and* $\mathbf{b}(1)$ *are extreme points of* $\operatorname{conv}\{\mathbf{b}_0, \ldots, \mathbf{b}_n\}$.

(iii) Subdivision of $\mathbf{b}(t)$ *at* $\alpha \in (0,1)$ *yields subcurves which are weakly bcp-increasing in direction* $\mathbf{m}$ *and whose enclosing polygons intersect only at* $\mathbf{b}(\alpha)$. *If* $\mathbf{b}(t)$ *is bcp-increasing, so are the subcurves.*

PROOF. *Ad (i).* It suffices to show $\langle \mathbf{m}, \mathbf{b}'(t) \rangle > 0$ for $t \in (0,1)$; monotonicity at 0 and 1 follows from continuity. We have $\mathbf{b}'(t) = n \sum_{i=0}^{n-1} \Delta\mathbf{b}_i B_i^{n-1}(t)$, and $B_i^{n-1}(t) > 0$ for $t \in (0,1)$. If $\Delta\mathbf{b}_i \neq 0$, then $\langle \mathbf{m}, \Delta\mathbf{b}_i \rangle > 0$. Hence all terms of $\langle \mathbf{m}, \mathbf{b}'(t) \rangle$ are non-negative, and there is at least one positive term, as $\mathbf{b}(t)$ is non-constant. If $\mathbf{b}(t)$ is bcp-increasing, we also have $\mathbf{b}'(0) = n\Delta\mathbf{b}_0 \neq 0$ and $\mathbf{b}'(1) = n\Delta\mathbf{b}_{n-1} \neq 0$.

Ad (ii). $\mathbf{b}_0$ and $\mathbf{b}_n$ are the unique minimizer and maximizer, resp., of $\langle \mathbf{m}, \cdot \rangle$ in the set $\{\mathbf{b}_0, \ldots, \mathbf{b}_n\}$.

Ad (iii). The two subcurves are $\mathbf{b}^-(t) = \sum_{i=0}^{n} \mathbf{b}_0^i B_i^n(t)$ and $\mathbf{b}^+(t) = \sum_{i=0}^{n} \mathbf{b}_i^{n-i} B_i^n(t)$. The concatenated left, lower, and right sides of a truncated de Casteljau triangle consisting only of rows $0, \ldots, j$ form a sequence

$$(\mathbf{b}_0^0, \ldots, \mathbf{b}_0^j, \ldots, \mathbf{b}_{n-j}^j, \ldots, \mathbf{b}_n^0).$$

For $j = 0$, this is the control polygon of $\mathbf{b}(t)$, hence two successive points are $\mathbf{m}$-increasing (or equal). This property is preserved by the de Casteljau algorithm as j increases and thus holds at $j = n$ for $(\mathbf{b}_0^0, \ldots, \mathbf{b}_0^n, \ldots, \mathbf{b}_n^0)$. Hence $\mathbf{b}^-(t)$ and $\mathbf{b}^+(t)$ are (weakly) bcp-increasing in direction $\mathbf{m}$. The common control point $\mathbf{b}_0^n$ is the unique maximizer of $\langle \mathbf{m}, \cdot \rangle$ in the set $\{\mathbf{b}_0^0, \ldots, \mathbf{b}_0^n\}$ and its unique minimizer in the set $\{\mathbf{b}_0^n, \ldots, \mathbf{b}_n^0\}$. This implies $\operatorname{conv}\{\mathbf{b}_0^0, \ldots, \mathbf{b}_0^n\} \cap \operatorname{conv}\{\mathbf{b}_0^n, \ldots, \mathbf{b}_n^0\} = \{\mathbf{b}_0^n\}$. □

We call a weakly bcp-monotone Bézier curve $\sum_{i=0}^{n} \mathbf{b}_i B_i^n(t)$ *straight* if $\mathbf{b}_0, \ldots, \mathbf{b}_n$ are collinear.

LEMMA 4. *Let* $\mathbf{b}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(t)$ *be a weakly bcp-monotone Bézier curve.*

(i) If $\mathbf{b}(t)$ *is straight,* $\mathbf{b}([0,1]) = \operatorname{conv}\{\mathbf{b}_i\}_i = \overline{\mathbf{b}(0)\,\mathbf{b}(1)}$.

(ii) If $\mathbf{b}(t)$ *is not straight,* $\mathbf{b}((0,1))$ *is contained in the interior of* $\operatorname{conv}\{\mathbf{b}_i\}_i$.

By Lemma 3(ii), the enclosing polygon f of $\mathbf{b}$ has $\mathbf{b}(0)$ and $\mathbf{b}(1)$ among its vertices. We will use f in place of $\mathbf{b}$ as an edge in a planar graph. This prompts the following definition: A *fat edge* is a convex polygon f, two of whose vertices are distinguished as *endpoints*, in our case: $\mathbf{b}(0)$ and $\mathbf{b}(1)$. A *fat planar graph* is an undirected graph (V, F) with a finite set $V \subseteq \mathbb{R}^2$ of *fat vertices* and a finite edge set F of fat edges with endpoints in V such that distinct fat edges intersect only in common endpoints, and a fat vertex intersects a fat edge only if it is an endpoint.

3. THE ALGORITHM

Consider the half-open *unit pixel* $U := [-\frac{1}{2}, +\frac{1}{2}) \times [-\frac{1}{2}, +\frac{1}{2})$. We partition $\mathbb{R}^2$ into *pixels* $\mathbf{a} + U$ around *centers* $\mathbf{a} \in \mathbb{Z}^2$. Each point $(x, y) \in \mathbb{R}^2$ is contained in a unique pixel with center $\rho((x,y)) = (\lfloor x + \frac{1}{2} \rfloor, \lfloor y + \frac{1}{2} \rfloor)$. A straight-line segment $\overline{\mathbf{pq}}$ *passes through* a pixel V if $V \cap \overline{\mathbf{pq}} \neq \emptyset$ and $V \cap \{\mathbf{p}, \mathbf{q}\} = \emptyset$.

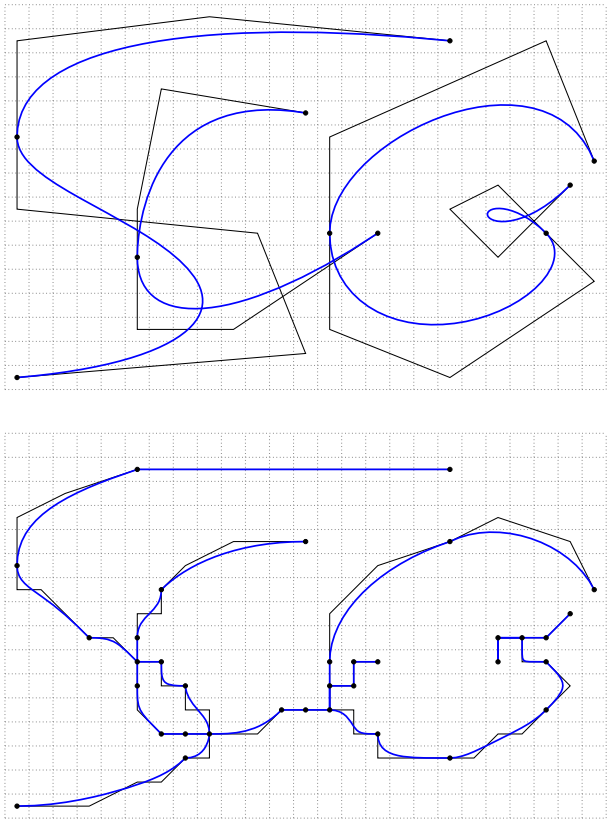

Figure 1: An example input consisting of seven intersecting Bézier curves that have integer control points and form the letters SCG (above). Computing their snap-rounded arrangement has resulted in 35 different fragments. For demonstration, we show the rounded arrangement that would result without merging (below): the two urcurves forming the top arcs of letters S and G are needlessly fragmented. With merging, they are the same in the output as in the input (see above).

As in straight-line snap rounding, we say that $\overline{\mathbf{pq}}$ and a point $\mathbf{r}$ *conflict* if $\overline{\mathbf{pq}}$ passes through the pixel around $\rho(\mathbf{r})$.

Let S be an input set of non-overlapping Bézier curves. We assume pixel boundaries to be in general position to S. In particular, we assume that intersections of curves, irregular points and self-intersections lie in the interiors of pixels, and that no curve intersects a pixel boundary tangentially or in a corner.

An implementation can overcome this restriction by adding a randomized fractional offset vector $\mathbf{v}$ to the integer grid. As all significant points have to lie on either side of the pixel boundary, but never on it, an implementation can choose the coordinates of $\mathbf{v}$ randomly bit after bit and never has to commit to an exact value. In expectancy, the required number of bits is logarithmic in the number of comparisons made. For ease of exposition, we do not consider this in the sequel and simply make the assumptions stated above.

Our algorithm receives as input the set S of Bézier curves, which we call *urcurves*. In a **preprocessing phase**, we subdivide urcurves into bcp-monotone pieces. In two further subdivision phases, we achieve that their enclosing poly-

gons form a fat planar graph T (**graph building phase**), and that there are no conflicts between control points and the straight-line segments linking them (**conflict removal phase**). In principle, we are then ready for rounding, but this would not yet produce good results, because our algorithm subdivides a lot; and often this merely serves to enclose a curve more tightly but is not required for rounding. Hence there is a **merging phase** for subcurves before they are rounded and output in the **rounding phase**. Figure 1 shows an example of avoidable fragmentation.

3.1 The preprocessing phase

If an urcurve does not have an irregular point, repeated subdivision will partition it into bcp-monotone pieces. (Proof: If $0 \notin \mathbf{b}'([0,1])$, repeated subdivision of $\mathbf{b}'(t)$ will refine its polygonal enclosure away from 0. Eventually, Lemma 1 applies to all subcurves.) But what if there is an irregular point? By our genericity assumption, this point lies in the interior of a pixel. Under repeated subdivision, it will end up on a subcurve that is *trivial*, meaning all its control points are in the same pixel. We can round a trivial curve to a single vertex, namely the center of the pixel containing it. Hence we can reduce it, already before rounding, to one of its endpoints (or both), which will round to a vertex at the pixel center and represent the irregular point.

So the preprocessing phase takes a set S, initially the set of urcurves, and as long as there is a curve $\mathbf{b} \in S$, $\mathbf{b}$ is extracted from S and examined. If $\mathbf{b}$ is bcp-monotone, $\mathbf{b}$ is added to the set Q. If $\mathbf{b}$ is not bcp-monotone but trivial, the endpoint $\mathbf{b}(0)$ is added to the set Q. Otherwise, $\mathbf{b}$ is subdivided and both subcurves are put back into S. When S has become empty, the preprocessing phase terminates with output Q.

3.2 The graph building phase

In this phase, we insert the preprocessed curves and their endpoints into a fat planar graph T. For each fat edge of T, we do not just store the curve $\mathbf{b}$ it stands for, but also the urcurve $ur[\mathbf{b}]$ of which it is a subcurve, and its parameter interval $[t^0, t^1]$ on the urcurve. For each fat vertex v of T, we store its *preimages*; that is, all pairs $(ur[\mathbf{b}], t_v)$ of an urcurve and a parameter $t_v \in [0,1]$ such that $ur[\mathbf{b}](t_v)$ is the point at which vertex v resides.

Graph building proceeds as follows:

While $Q \neq \emptyset$, we extract an element from it and insert it into T. We need to distinguish between insertion of points and of curves, and we need to process curves that are *relegated* from T in the process.

To insert a point $\mathbf{p}$ as a fat vertex, we first check whether there is an *obstacle* to its insertion, i.e., a fat edge f in T such that $\mathbf{p} \in f$ but $\mathbf{p}$ is not an endpoint of f. If so, we remove f from T and relegate the curve it stands for (see below). Finally, we insert $\mathbf{p}$ into T.

To insert a curve $\mathbf{b}$ as a fat edge, we first insert its endpoints in the manner just described. Then we check whether there is an *obstacle* to the insertion of $\mathrm{conv}\{\mathbf{b}_i\}_i$, i.e., a fat vertex v or a fat edge f that intersects $\mathrm{conv}\{\mathbf{b}_i\}_i$ in a point other than an endpoint. If so, we relegate $\mathbf{b}$; and if the obstacle is a fat edge f, we remove f from T and also relegate the curve it stands for. If we found no obstacle, we insert $\mathrm{conv}\{\mathbf{b}_i\}_i$ as fat edge from $\mathbf{b}(0)$ to $\mathbf{b}(1)$.

For each relegated curve, we check whether it is trivial. If it is not, we subdivide it and re-insert its parts into Q.

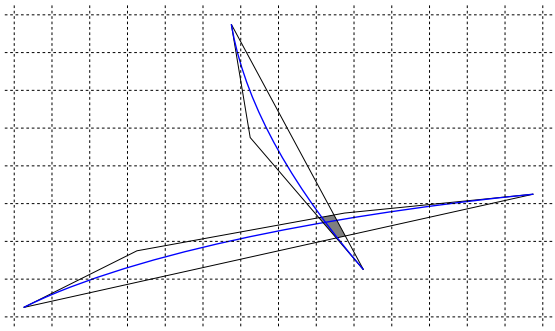

Figure 2: Two curves with enclosing polygons that intersect in a single pixel.

If two curves were relegated because their enclosing polygons intersect, and if this intersection lies inside a single pixel V (as depicted in Figure 2), then, as an optimization, one can split both curves at V, discard the small intersecting pieces inside V, and attain non-intersecting subcurves ending at V. This reproduces the method from CAGD practice [21, §3.7] for intersection of long flat curves in our framework, with "inside one pixel" in place of "error $< \varepsilon$". It improves on repeated subdivision at $\alpha = \frac{1}{2}$ by replacing binary search with direct computation.

3.3 The conflict removal phase

In this phase, we *subdivide edges* of T. That means, we remove an edge, subdivide the curve it stands for, and then re-insert the enclosing polygons of the two resulting subcurves into T. This re-insertion never faces an obstacle in T: The two new fat edges are subsets of the removed fat edge, so no other fat edge is an obstacle to their insertion, and by Lemma 3(iii), they are not obstacles to each other.

We subdivide edges for two reasons. One reason is to ensure that each curve remains weakly bcp-monotone when rounded. The primary reason, however, is to remove conflicts between points and segments in the sense of straight-line snap rounding: A control point $\mathbf{r}$ of some curve must not conflict with a straight-line segment $\overline{\mathbf{pq}}$ between two control points of another curve; otherwise, $\rho(\mathbf{r})$ might end up on the wrong side of $\overline{\rho(\mathbf{p})\,\rho(\mathbf{q})}$. For this, we need to consider the whole *control clique*, or *CC*, of a Bézier curve, that is the graph of all control points and all straight-line segments between them.

Conflict removal proceeds as follows:

If there exists a fat edge f and a fat vertex v that is not an endpoint of f such that v conflicts with a CC edge of the curve represented by f, then we subdivide f and reiterate.

If there exist two distinct fat edges such that a CC vertex of one conflicts with a CC edge of the other, we subdivide both and reiterate.

If there exists a non-trivial fat edge f representing a Bézier curve $\sum_i \mathbf{b}_i B_i^n(t)$ such that its rounding $\sum_i \rho(\mathbf{b}_i) B_i^n(t)$ is not weakly bcp-monotone, then we subdivide f and reiterate.

If none of the three conditions applies, the conflict removal phase terminates. This happens in the worst case when every urcurve has been subdivided into pieces that are either trivial or have a weakly monotone control polygon $(\mathbf{p}, \ldots, \mathbf{p}, \mathbf{q}, \ldots, \mathbf{q})$ that connects two adjacent pixels.

All we have done so far is to subdivide urcurves and to and discard some trivial subcurves (except at least one endpoint of each). We call the remaining subcurves the *unrounded fragments*.

PROPOSITION 5. *Let* $\mathbf{b}$ *be a non-trivial unrounded fragment. Let* $V = \rho(\mathbf{b}_{i_0}) + U$ *be a pixel that contains a control point of* $\mathbf{b}$ *and a control point* $\mathbf{q}$ *of another unrounded fragment. Then* $\operatorname{conv}\{\mathbf{b}_i\}_i \cap \partial V$ *is connected.*

The symbol ∂ denotes the boundary of a point set. The proposition follows directly from the next lemma.

LEMMA 6. *With notation as above,* $\operatorname{conv}\{\mathbf{b}_i\}_i \setminus V$ *is connected.*

PROOF. We begin by showing: any connected component C_1 of $\operatorname{conv}\{\mathbf{b}_i\}_i \setminus V$ contains a control point $\mathbf{b}_{j_1}$. Take an arbitrary point $\mathbf{p} \in C_1$. As $\mathbf{p} \notin V$, there exists a half-plane H that contains $\mathbf{p}$ but is disjoint from V. As the complementary half-plane $\mathbb{R}^2 \setminus H$ is convex but does not contain the element $\mathbf{p}$ of $\operatorname{conv}\{\mathbf{b}_i\}_i$, it cannot contain all points $\mathbf{b}_i$. Hence there is $\mathbf{b}_{j_1} \in H$; it remains to show $\mathbf{b}_{j_1} \in C_1$. By construction, the set $\operatorname{conv}\{\mathbf{b}_i\}_i \cap H$ is a convex and thus connected subset of $\operatorname{conv}\{\mathbf{b}_i\}_i \setminus V$, and so lies entirely within one connected component. Since $\mathbf{p} \in \operatorname{conv}\{\mathbf{b}_i\}_i \cap H$, we have $\operatorname{conv}\{\mathbf{b}_i\}_i \cap H \subseteq C_1$ and so $\mathbf{b}_{j_1} \in C_1$.

To prove the lemma, assume for contradiction that there is a connected component $C_2 \neq C_1$. It contains a control point $\mathbf{b}_{j_2}$, giving rise to the segment $\overline{\mathbf{b}_{j_1}\mathbf{b}_{j_2}}$, which is a subset of $\operatorname{conv}\{\mathbf{b}_i\}_i$ but, by disconnectedness, not of $\operatorname{conv}\{\mathbf{b}_i\}_i \setminus V$. Hence $\overline{\mathbf{b}_{j_1}\mathbf{b}_{j_2}}$ conflicts with $\mathbf{q}$ at V, a contradiction. □

3.4 The merging phase

We will now reduce fragmentation by merging certain chains of unrounded fragments. A *chain* is a sequence $e = (\mathbf{b}^1, \dots, \mathbf{b}^k)$ of fragments that have a common urcurve $\mathbf{b} = ur[\mathbf{b}^j]$ and whose parameter intervals $[t^0[\mathbf{b}^j], t^1[\mathbf{b}^j]]$ on $\mathbf{b}$ are adjacent, that is, $t^1[\mathbf{b}^j] = t^0[\mathbf{b}^{j+1}]$. We call a chain *trivial* if all fragments on it are trivial. We can regard the chain e as one subcurve $\mathbf{c}^e(t) = \sum_{i=0}^n \mathbf{c}_i^e B_i^n(t)$ of $\mathbf{b}$ that corresponds to the parameter interval $I^e := [t^0[\mathbf{b}^1], t^1[\mathbf{b}^k]]$ of $\mathbf{b}$. If the chain e is non-trivial, so is the curve $\mathbf{c}^e$. The control points of each unrounded fragment $\mathbf{b}^j$ on $\mathbf{c}^e$ are convex combinations $\mathbf{b}_i^j = \sum_{\ell=0}^n \mu_\ell \mathbf{c}_\ell^e$, with coefficients μ_ℓ that depend only on the parameter subinterval of $\mathbf{b}^j$ on $\mathbf{c}^e$, not on the location of control points. So when we round $\mathbf{c}^e(t) = \sum_{i=0}^n \mathbf{c}_i^e B_i^n(t)$ to $\rho\mathbf{c}^e(t) = \sum_{i=0}^n \rho(\mathbf{c}_i^e) B_i^n(t)$, this implicitly moves $\mathbf{b}_i^j = \sum_{\ell=0}^n \mu_\ell \mathbf{c}_\ell^e$ to $\tilde{\mathbf{b}}_i^j = \sum_{\ell=0}^n \mu_\ell \rho(\mathbf{c}_\ell^e)$. Except for $\mathbf{b}_0^1 = \mathbf{c}^e(0)$ and $\mathbf{b}_n^k = \mathbf{c}^e(1)$, all control points $\mathbf{b}_i^j$ are rounded to locations $\tilde{\mathbf{b}}_i^j$, which, in general, are not pixel centers. This requires new ideas to rule out inversion of topology during rounding.

The first idea is this: The displacements $\tilde{\mathbf{b}}_i^j - \mathbf{b}_i^j$ are elements of the inverted unit pixel $-U$, because $\rho(\mathbf{c}_i^e) - \mathbf{c}_i^e$ is an element of the convex set $-U$ for all i. We define the *pixel neighbourhood* $N(\mathbf{p})$ of a point $\mathbf{p}$ to be the pixel around $\rho(\mathbf{p})$ and its eight neighbours; that is, $N(\mathbf{p}) := (\rho(\mathbf{p}) + \{-1, 0, +1\}^2) + U$. Objects outside $N(\mathbf{b}_i^j)$ cannot collide with $\mathbf{b}_i^j$ during rounding, as we will see in the proof of Lemma 11.

The second idea addresses objects inside $N(\mathbf{b}_i^j)$, provided that they have the same urcurve $\mathbf{b}$. This is to be expected for nearby fragments on the same chain. We will build a subcurve of $\mathbf{b}$ comprising everything that is in the pixel neighbourhood of a control point $\mathbf{b}_i^j$ and make sure that this curve is bcp-monotone after rounding; then Lemma 3(iii) yields separation of the rounded fragments. To guarantee bcp-monotonicity after rounding, we modify the criterion for bcp-monotonicity from Lemma 1 by introducing some slack: We say that a Bézier curve $\mathbf{b}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ is *λ-robustly bcp-monotone* if $\lambda > 0$ and

$$0 \notin \operatorname{int} \operatorname{conv}\left(\{\Delta\mathbf{b}_0, \dots, \Delta\mathbf{b}_{n-1}\} + \lambda\{-1, +1\}^2\right). \qquad (1)$$

If $\mathbf{b}$ is bcp-monotone, there exists a maximal value of λ that satisfies (1), and we call it the *robustness number* of $\mathbf{b}$. It indicates how much larger a curve $\mathbf{c}$ has to be compared to its subcurve $\mathbf{b}$ so that the bcp-monotonicity of $\mathbf{b}$ is not destroyed when the control points of $\mathbf{c}$ are translated by vectors from $-U$.

LEMMA 7. *Consider* $\mathbf{c}(t) = \sum_{i=0}^n \mathbf{c}_i B_i^n(t)$ *and its subcurve* $\mathbf{b}(t) = \mathbf{c}((1-t)a + tb)$ *with* $0 \le a < b \le 1$. *Let* $\mathbf{b}$ *be* $(b-a)$*-robustly bcp-monotone. Then there exists a unit-length vector* $\mathbf{m}$ *with the following property:*
For all perturbed curves $\tilde{\mathbf{c}}(t) = \sum_{i=0}^n \tilde{\mathbf{c}}_i B_i^n(t)$ *that satisfy the error bound* $\mathbf{e}_i := \tilde{\mathbf{c}}_i - \mathbf{c}_i \in -U$ *for* $i = 0, \dots, n$, *the subcurve* $\tilde{\mathbf{b}}(t) = \tilde{\mathbf{c}}((1-t)a + tb)$ *is bcp-increasing in direction* $\mathbf{m}$.

PROOF. Let C be the set on the right-hand side of (1). We have $0 \notin C$, so there is a half-plane $\langle \mathbf{m}, \cdot \rangle > 0$ containing C. The lemma claims $\langle \mathbf{m}, \Delta\tilde{\mathbf{b}}_i \rangle > 0$ for all i. To prove this, we show $\Delta\tilde{\mathbf{b}}_i - \Delta\mathbf{b}_i \in (b-a)(-1, +1)^2$ and thus $\Delta\tilde{\mathbf{b}}_i \in C$. To do so, we invoke the theory of *blossoms* [22] [21, 3.1] [4, 4.7]. There exists a unique symmetric n-affine form $\mathbf{c}[t_1, \dots, t_n]$ with $\mathbf{c}(t) = \mathbf{c}[t, \dots, t]$. The control points of $\mathbf{b}$ are $\mathbf{b}_i = \mathbf{c}[a, \dots, a, b, \dots, b]$, where exactly i arguments are b. From $t = t \cdot 1 + (1-t) \cdot 0$ it follows that

$$\mathbf{c}[\dots t \dots] = t\mathbf{c}[\dots 1 \dots] + (1-t)\mathbf{c}[\dots 0 \dots];$$

hence

$$\begin{aligned} \Delta\mathbf{b}_i &= \mathbf{c}[\dots b \dots] - \mathbf{c}[\dots a \dots] \\ &= (b-a)(\mathbf{c}[\dots 1 \dots] - \mathbf{c}[\dots 0 \dots]) \end{aligned}$$

and so

$$\begin{aligned} \Delta\tilde{\mathbf{b}}_i - \Delta\mathbf{b}_i &= (b-a)(\mathbf{e}[\dots 1 \dots] - \mathbf{e}[\dots 0 \dots]) \\ &\in (b-a)(-U + U) = (b-a)(-1, +1)^2, \end{aligned}$$

as desired. □

Now we give conditions (M1–3) under which a chain $e = (\mathbf{b}^1, \dots, \mathbf{b}^k)$ with urcurve $\mathbf{b}$ can be merged. The fat vertices at $\mathbf{b}_0^1$ and $\mathbf{b}_n^k$ are the *endpoints* of e. The fat vertices at $\mathbf{b}_0^2, \dots, \mathbf{b}_0^k$ are *fat inner vertices* of e. The remaining control points $\mathbf{b}_i^j$ $(1 \le j \le k,\ 0 < i < n)$ are *thin inner vertices* of e.

(M1) An inner fat vertex v of e has only one preimage $(\mathbf{b}, t_v)$ (see §3.2) and exactly two incident fat edges. (Necessarily, these two fat edges are the two unrounded fragments of e incident to v, and t_v is the common boundary point of their parameter intervals on $\mathbf{b}$.)

For a point $\mathbf{p}$, we let $\mathcal{N}(\mathbf{p})$ denote the set comprising all fat vertices v of T located in $N(\mathbf{p}) \setminus \{\mathbf{p}\}$, and comprising all fat edges f of T such that $N(\mathbf{p})$ intersects an edge of $CC(f)$ which does not have $\mathbf{p}$ as an endpoint.

(M2) For an inner vertex v of e, all elements of $\mathcal{N}(v)$ have the same urcurve $\mathbf{b}$ as the chain e.

Suppose (M2) holds. If v is an inner fat vertex, let I^v be the smallest subinterval of $[0,1]$ that contains the parameter values of $\mathbf{b}$ for the elements of $\mathcal{N}(v) \cup \{v\}$; or formally, $I^v = \mathrm{conv}(\mathbf{b}^{-1}(\mathcal{N}(v) \cup \{v\}))$. Likewise, if v is an inner thin vertex of e belonging to fat edge f, let $I^v = \mathrm{conv}(\mathbf{b}^{-1}(\mathcal{N}(v) \cup \{f\}))$. In either case, denote by $\mathbf{h}^v$ the subcurve of $\mathbf{b}$ corresponding to I^v. We call $\mathbf{h}^v$ the *neighbourhood curve* of v. In case v is an inner fat vertex with $\mathcal{N}(v) = \emptyset$, I^v has only one element. Otherwise, $\mathbf{h}^v$ is non-constant, because $\mathbf{b}$ is non-constant. The control points of $\mathbf{h}^v$ are convex combinations of $\mathbf{c}_0, \ldots, \mathbf{c}_n$ with fixed coefficients, so going from $\mathbf{c}$ to $\rho\mathbf{c}$ turns $\mathbf{h}^v$ into the rounded neighbourhood curve $\tilde{\mathbf{h}}^v$, as for fragments. Let $\lambda_{\mathbf{h}^v}$ be the robustness number of $\mathbf{h}^v$, or 0 if $\mathbf{h}^v$ is constant. Recall that I^e is the parameter interval of $\mathbf{c}^e$ on $\mathbf{b}$.

(M3) For an inner vertex v of e, it holds that $I^v \subseteq I^e$ and $|I^v|/|I^e| \leq \lambda_{\mathbf{h}^v}$.

We call a chain e *admissible* if it satisfies (M1–3) or consists of just one fragment. Given the fat graph T as resulting from the conflict removal phase, the merging phase computes admissible chains of maximal length by first forming maximal candidate chains satisfying (M1) and then breaking up candidate chains iteratively until (M2) and (M3) are also satisfied or until the length has dropped to 1. Trivial chains are discarded (but not their endpoints). The curves $\mathbf{c}^e$ for all non-trivial maximal admissible chains e are called the *unrounded meta-fragments*. Each unrounded meta-fragment $\mathbf{c}^e$ arises either from a chain of length $k = 1$, in which case we call $\mathbf{c}^e$ a *singleton*, or from a chain of length $k \geq 2$ that satisfies (M1–3). Let E be the set of all unrounded meta-fragments. Let V^* be the set of all fat vertices of T except the inner fat vertices of unrounded meta-fragments. The output of the merging phase is the graph (V^*, E). It is the unrounded form of the arrangement that we compute.

Observe that every non-trivial fragment appears in exactly one meta-fragment; every trivial fragment appears in at most one meta-fragment. If a fragment appears in a non-singleton meta-fragment, we call it *merged*, otherwise *unmerged*. If $\mathbf{p}$ is a control point of a merged fragment and is not the endpoint of its meta-fragment, we call $\mathbf{p}$ *merged*, otherwise *unmerged*. We say two unrounded fragments *have a common neighbourhood curve* $\mathbf{h}$, if they belong to the same meta-fragment $\mathbf{c}$, and $\mathbf{c}$ has an inner vertex v such that $\mathbf{h} = \mathbf{h}^v$ comprises both fragments. By (M3) in conjunction with Lemma 7, its rounding $\tilde{\mathbf{h}}$ is bcp-monotone.

3.5 The rounding phase

We obtain a snap rounding $\mathcal{A}'$ of the arrangement $\mathcal{A}$ induced by the set S of urcurves as follows. The vertex set of $\mathcal{A}'$ is $V' := \rho V^* = \{\rho(v) \mid v \in V^*\}$. This unites previously distinct vertices within a pixel. The edge set is $E' := \rho E := \{\rho\mathbf{c}^e \mid e \in E\}$, whose elements we call *rounded meta-fragments*. Each rounded meta-fragment $\rho\mathbf{c}$ is computed from a corresponding unrounded meta-fragment $\mathbf{c}$ by just rounding each control point $\mathbf{c}_i$ to $\rho(\mathbf{c}_i)$. (As $\mathbf{c}$ is non-trivial, $\rho\mathbf{c}$ is non-constant.) In our graph, we regard $\rho\mathbf{c}$ as an edge from vertex $\rho\mathbf{c}(0)$ to vertex $\rho\mathbf{c}(1)$. That concludes the algorithm. We analyze the properties of $\mathcal{A}'$ in the next section.

4. GEOMETRIC ANALYSIS

To what extent does the rounded arrangement $\mathcal{A}'$ reflect the geometry of the true arrangement $\mathcal{A}$?

THEOREM 8. *Let $\mathbf{c}$ be an unrounded meta-fragment. The rounded meta-fragment $\rho\mathbf{c}$ is contained in the* sausage region *$\mathbf{c}([0,1]) + (-U)$ of $\mathbf{c}$, and thus also in the sausage region of the urcurve $\mathbf{b}$ of $\mathbf{c}$. Conversely, let $\mathbf{b}$ be an urcurve. Then there exists a subset $F_{\mathbf{b}}$ of $V' \cup E'$ such that $\mathbf{b}([0,1])$ is contained in $(\bigcup F_{\mathbf{b}}) + U$. Finally, let $\mathbf{p}$ be an endpoint of an urcurve, an irregular point of an urcurve, a self-intersection point of an urcurve, or an intersection point of two urcurves. Then $\rho(\mathbf{p})$ is a vertex in V'.*

PROOF. Consider $\mathbf{c}$. For any $t \in [0,1]$ we have $(\rho\mathbf{c})(t) - \mathbf{c}(t) = \sum_{i=0}^{n} (\rho(\mathbf{c}_i) - \mathbf{c}_i) B_i^n(t) \in -U$, The claim on $\mathbf{b}$ is immediate from $\mathbf{c}([0,1]) \subseteq \mathbf{b}([0,1])$. The converse inclusion holds, because the only subcurves $\mathbf{b}^*$ of $\mathbf{b}$ that we ever delete are trivial, but we retain the endpoint $\mathbf{b}^*(0)$ as a non-mergeable fat vertex, and we have $\mathbf{b}^*([0,1]) \subseteq \rho(\mathbf{b}^1(0)) + U$. The statement on $\mathbf{p}$ is easily verified by inspection of the algorithm. □

So much for approximate preservation of metric properties; now we turn to the topological properties of $\mathcal{A}'$. We generalize the approach of [8] from independent straight-line segments to our setting with enclosing polygons: We view rounding as a deformation $\mathcal{D}_s$ continuous in time $s \in [0,1]$ that interpolates between "zero snaproundedness" at $s = 0$ and "full snaproundedness" at $s = 1$. $\mathcal{D}_s$ deforms the control cliques of unrounded fragments to the control cliques of rounded fragments.[2] To define $\mathcal{D}_s$, we need the notion of a hot pixel. A pixel V is *hot* if there exist at least two different unrounded fragments that have an unmerged control point in V.

LEMMA 9. *Let V be a hot pixel.*
(i) No control clique edge passes through V.
(ii) All control points in V are unmerged and thus round to the pixel center.

PROOF. There exist two different unrounded fragments $\mathbf{b}^j$, $j \in \{1,2\}$, and for each of them an unmerged control point $\mathbf{r}_j \in V$.

Ad (i). Suppose some CC edge $\overline{\mathbf{pq}}$ belonging to an unrounded fragment $\mathbf{b}^3$ passes through V. After the conflict removal phase, such a conflict between $\overline{\mathbf{pq}}$ and $\mathbf{r}_j$, $j \in \{1,2\}$, is only possible if $\mathbf{b}^j = \mathbf{b}^3$, but this cannot hold for $j = 1$ and $j = 2$ simultaneously. Hence such $\overline{\mathbf{pq}}$ cannot exist.

Ad (ii). Assume there is a merged control point $\mathbf{p}$ of some unrounded fragment $\mathbf{b}^3$ in V. Its pixel neighbourhood $N(\mathbf{p})$ intersects the enclosing polygons of $\mathbf{b}^1$ and $\mathbf{b}^2$. (M2) and (M3) imply that $\mathbf{b}^1$, $\mathbf{b}^2$, and $\mathbf{b}^3$ belong to the same unrounded meta-fragment $\mathbf{c}$, and that the neighbourhood curve $\mathbf{h}$ of $\mathbf{p}$ comprises $\mathbf{b}^1$ and $\mathbf{b}^2$. As $\mathbf{r}_1$ and $\mathbf{r}_2$ are unmerged control points, they must be the two endpoints of the meta-fragment $\mathbf{c}$. It follows that $\mathbf{h} = \mathbf{c}$. After rounding, the unmerged control points $\rho\mathbf{c}(0) = \tilde{\mathbf{h}}(0)$ and $\rho\mathbf{c}(1) = \tilde{\mathbf{h}}(1)$ coincide in the pixel center, contradicting the bcp-monotonicity of $\tilde{\mathbf{h}}$. Hence such $\mathbf{p}$ cannot exist. □

This lemma does not hold for a pixel V with just one control point of one curve, hence our new definition of "hot".

[2]This notion of "deformation" matches [8]. It is not a deformation of the entire plane as in [18].

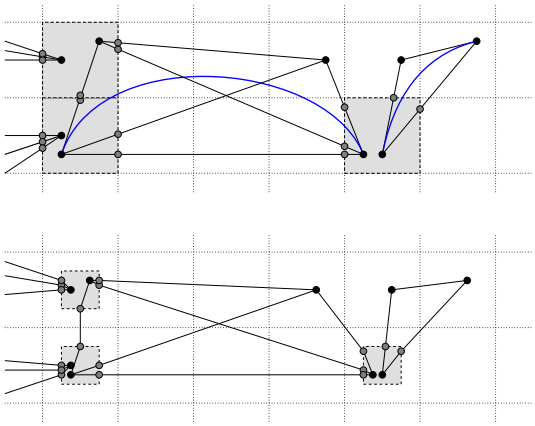

Figure 3: extended control cliques under the deformation $\mathcal{D}_s$ at $s = 0$ (above) and $s = \frac{1}{2}$ (below).

Consider the control clique of an unrounded fragment $\mathbf{b}^j$. For every hot pixel V, we split a CC edge $\overline{\mathbf{pq}}$ by introducing an additional vertex $\mathbf{a}$ wherever the edge intersects the pixel boundary ∂V, and we assign the center of V as rounded position $\tilde{\mathbf{a}}$ to the vertex $\mathbf{a}$. (If $\overline{\mathbf{pq}}$ intersects the common boundary line segment of two adjacent hot pixels, this means we introduce two new vertices at that point, one belonging to each hot pixel.) The result of this we call *extended control clique*, or *ECC*. There are three kinds of vertices in an ECC: the *boundary vertices* which we just introduced; the *internal vertices* inside and the *external vertices* outside hot pixels. Likewise, there are two kinds of ECC edges: *internal edges* within hot pixels and *external edges* outside hot pixels. By Lemma 9(i), all internal edges have at least one internal vertex among their endpoints.

Let us now define how $\mathcal{D}_s$ acts on ECCs. Each ECC vertex $\mathbf{a}$ has been assigned a rounded position $\tilde{\mathbf{a}}$. (For control points this was done in §3.4.) $\mathcal{D}_s$ interpolates ECC vertices linearly: $\mathcal{D}_s(\mathbf{a}) = (1-s)\mathbf{a} + s\tilde{\mathbf{a}}$. We have $\mathcal{D}_s(\mathbf{p}) - \mathbf{p} \in s(-\frac{1}{2}, +\frac{1}{2}]^2 = -s \cdot U$ for internal and external ECC vertices (i.e., fragment control points) and $\mathcal{D}_s(\mathbf{a}) - \mathbf{a} \in s[-\frac{1}{2}, +\frac{1}{2}]^2$ for boundary ECC vertices. For an ECC edge $\overline{\mathbf{pq}}$, we define $\mathcal{D}_s(\overline{\mathbf{pq}}) = \overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$. See Figure 3 for an example.

At $s = 1$, all ECC boundary vertices and internal edges collapse with the respective internal vertices at pixel centers. Thus, the deformed ECC coincides with the control clique of the rounded fragment.

For any $s \in [0,1]$, the deformed ECC is a graph in the plane; its edges are straight-line segments that may cross. If we imagine further vertices at crossing points (*crossing vertices*), we have a planar graph with exactly one unbounded face. The union of all vertices, edges, and bounded faces is a closed polygon, which we call *ECC polygon* $\mathcal{P}_s(\mathbf{b}^j)$ of the fragment $\mathbf{b}^j$ at time s. We have $\mathcal{P}_0(\mathbf{b}^j) = \operatorname{conv}\{\mathbf{b}_i^j\}_i$ and $\mathcal{P}_1(\mathbf{b}^j) = \operatorname{conv}\{\tilde{\mathbf{b}}_i^j\}_i$, so that the ECC polygon encloses the fragment at $s = 0$ and $s = 1$, respectively. This is the link between the ECC polygons and curves. However, we have no control over curves defined by control points $\mathcal{D}_s(\mathbf{b}_i^j)$ for $0 < s < 1$, because we cannot break Bézier control polygons at the boundaries of hot pixels. Also, $\mathcal{P}_s(\mathbf{b}^j)$ is not necessarily convex for $0 < s < 1$.

How do deformed ECC polygons interact? We begin with the case of a common neighbourhood curve.

Proposition 10. *Let $\mathbf{c}$ be an unrounded meta-fragment of degree n that consists of fragments $\mathbf{b}^1, \dots, \mathbf{b}^k$. Consider indices $1 \le j_1 < j_2 \le k$ such that $\mathbf{b}^{j_1}$ and $\mathbf{b}^{j_2}$ have a common neighbourhood curve $\mathbf{h}$.*

(i) If $j_2 > j_1 + 1$, then $\forall s \in [0,1]\colon \mathcal{P}_s(\mathbf{b}^{j_1}) \cap \mathcal{P}_s(\mathbf{b}^{j_2}) = \emptyset$.

(ii) If $j_2 = j_1 + 1$, then $\forall s \in [0,1]\colon \mathcal{P}_s(\mathbf{b}^{j_1}) \cap \mathcal{P}_s(\mathbf{b}^{j_2}) = \{\mathcal{D}_s(\mathbf{b}_n^{j_1})\} = \{\mathcal{D}_s(\mathbf{b}_0^{j_2})\}$.

Proof. We show (ii); the proof of (i) is similar and simpler. Overloading notation to include the control points of $\mathbf{h}$, we let $\mathcal{D}_s(\mathbf{h}_i) = (1-s)\mathbf{h}_i + s\tilde{\mathbf{h}}_i$. By (M3) and Lemma 7, there exists $\mathbf{m}$ such that $\mathcal{D}_s(\mathbf{h}_0), \dots, \mathcal{D}_s(\mathbf{h}_n)$ is $\mathbf{m}$-increasing for all $s \in [0,1]$. By Lemma 3(iii), also $\mathcal{D}_s(\mathbf{b}_0^{j_1})$, $\mathcal{D}_s(\mathbf{b}_1^{j_1})$, $\dots$, $\mathcal{D}_s(\mathbf{b}_{n-1}^{j_1})$, $\mathcal{D}_s(\mathbf{b}_n^{j_1}) = \mathcal{D}_s(\mathbf{b}_0^{j_2})$, $\mathcal{D}_s(\mathbf{b}_1^{j_2})$, $\dots$, $\mathcal{D}_s(\mathbf{b}_n^{j_2})$ is $\mathbf{m}$-increasing. Thus, the line $\ell_s(\mathbf{x}) := \langle \mathbf{m}, \mathbf{x} - \mathcal{D}_s(\mathbf{b}_n^{j_1}) \rangle$ separates the internal and external ECC vertices of $\mathbf{b}^{j_1}$ from those of $\mathbf{b}^{j_2}$, except for $\mathcal{D}_s(\mathbf{b}_n^{j_1}) = \mathcal{D}_s(\mathbf{b}_0^{j_2})$. Likewise, ℓ_s separates the boundary ECC vertices: This holds for $s = 0$ and $s = 1$, because then the boundary vertices are convex combinations of CC edge endpoints. So if $\mathbf{a}$ is any boundary ECC vertex, the linear expression $\ell_s(\mathcal{D}_s(\mathbf{a}))$ has the same sign at $s = 0$ and $s = 1$ and hence at any $s \in [0,1]$. It follows that the half-planes $\ell_s(\cdot) \lesseqgtr 0$ contain one ECC polygon each, except for the common vertex $\mathcal{D}_s(\mathbf{b}_n^{j_1})$ on the boundary line. □

Now we look at fragments without a common neighbourhood curve.

Lemma 11 (Main Lemma). *Consider two unrounded fragments $\mathbf{b}$ and $\mathbf{b}^*$ that do not have a common neighbourhood curve. Let $\overline{\mathbf{pq}}$ be an ECC edge of $\mathbf{b}$ and $\mathbf{r}$ an ECC vertex of $\mathbf{b}^*$ such that $\mathbf{r} \notin \{\mathbf{p}, \mathbf{q}\}$.*

(i) $\forall s \in [0,1)\colon \mathcal{D}_s(\mathbf{r}) \notin \overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$.

(ii) $\mathcal{D}_1(\mathbf{r}) \notin \overline{\mathcal{D}_1(\mathbf{p})\,\mathcal{D}_1(\mathbf{q})} \;\vee\; \mathcal{D}_1(\mathbf{r}) \in \{\mathcal{D}_1(\mathbf{p}), \mathcal{D}_1(\mathbf{q})\}$. If $\mathbf{r}$ is an external ECC vertex, the second alternative is excluded.

Proof. We begin with an observation on a hot pixel $\mathbf{a}+U$ ($\mathbf{a} \in \mathbb{Z}^2$). Consider the squares $Q_s = \mathbf{a} + (1-s)[-\frac{1}{2}, +\frac{1}{2}]^2$. Q_0 is the closure of the hot pixel. By Lemma 9(ii), all ECC vertices in it round to $\mathbf{a}$. Thus, for all ECC vertices and edges in Q_0, $\mathcal{D}_s$ acts as the affine shrinking map $Q_0 \to Q_s$. For $s < 1$, this map is bijective, but for $s = 1$, the entire pixel collapses to $Q_1 = \{\mathbf{a}\}$. Let $\mathbf{u}$ be any point on any ECC edge $\overline{\mathbf{vw}}$ outside Q_0. There is a line ℓ comprising a boundary edge of Q_0 that separates $\mathbf{u}$ from the pixel. We assume w.l.o.g. that it is the left edge of Q_0. Then ℓ equals ℓ_0, where $\ell_s := \mathbf{a} + (\{-\frac{1}{2}(1-s)\} \times \mathbb{R})$. The speed vector $\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{D}_s(\mathbf{u})$ is a convex combination of $\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{D}_s(\mathbf{v})$, $\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{D}_s(\mathbf{w})$ and thus an element of $[-\frac{1}{2}, +\frac{1}{2}]^2$. However, the line ℓ_s moves right with speed $\frac{1}{2}$, so for all $s \in [0,1]$, the point $\mathcal{D}_s(\mathbf{u})$ remains left and Q_s remains right of or on ℓ_s.

We have thus shown: *No point on an ECC edge, in particular no ECC vertex, moves from the outside into a shrinking hot pixel during the deformation $\mathcal{D}$.* We can now prove the lemma by case distinction.

Case 1: $\overline{\mathbf{pq}}$ is an internal edge. If $\mathbf{r}$ lies outside the hot pixel comprising $\overline{\mathbf{pq}}$, it never enters that pixel and thus cannot become a point of $\overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ for any s. If $\mathbf{r}$ lies inside that hot pixel or on its boundary, then initially $\mathbf{r} \notin \overline{\mathbf{pq}}$, because $\mathbf{r}$ is neither an endpoint (by premise) nor any other point (by construction). Hence $\mathcal{D}_s(\mathbf{r}) \notin \overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ for $s < 1$, proving (i); and $\{\mathcal{D}_1(\mathbf{r})\} = \overline{\mathcal{D}_1(\mathbf{p})\,\mathcal{D}_1(\mathbf{q})}$, proving (ii).

Case 2: $\overline{\mathbf{pq}}$ *is an external edge.* First, let $\mathbf{r}$ be an internal or boundary ECC vertex of some hot pixel Q_0. The points of $\overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ stay outside the shrinking hot pixel Q_s, with the potential exception of an endpoint $\mathcal{D}_s(\mathbf{p})$ that is a boundary ECC vertex of Q_s and coincides with $\mathcal{D}_s(\mathbf{r})$ in the pixel center at $s = 1$.

Next, let $\mathbf{r}$ be an unmerged external ECC vertex. We have $\mathcal{D}_1(\mathbf{r}) = \rho(\mathbf{r})$, and we know from conflict removal that $\overline{\mathbf{pq}}$ does not intersect $\rho(\mathbf{r}) + U$. In complete analogy to the argument above, we see that $\overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ does not enter the shrinking pixel containing $\mathcal{D}_s(\mathbf{r})$, so $\mathcal{D}_s(\mathbf{r}) \notin \overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ for all $s \in [0, 1]$.

Finally, let $\mathbf{r}$ be a merged external ECC vertex. It moves from $\mathcal{D}_0(\mathbf{r}) = \mathbf{r} \in \rho(\mathbf{r}) + U$ along $\mathcal{D}_s(\mathbf{r}) \in \mathbf{r} + (-s) \cdot U \subseteq \rho(\mathbf{r}) + (U + (-U)) = \rho(\mathbf{r}) + (-1, +1)^2$. Since $\mathbf{b}$ and $\mathbf{b}^*$ have no common neighbourhood curve, $\overline{\mathbf{pq}}$ does not intersect $N(\mathbf{r}) = \rho(\mathbf{r}) + [-\frac{3}{2}, +\frac{3}{2})^2$. As s grows, no point on $\overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})}$ changes by more than $\frac{1}{2}$ in any coordinate. Hence $\overline{\mathcal{D}_s(\mathbf{p})\,\mathcal{D}_s(\mathbf{q})} \cap (\rho(\mathbf{r}) + (-1, +1)^2) = \emptyset$ for all s, proving the claim. □

PROPOSITION 12. *Let the two unrounded fragments* $\mathbf{b}$ *and* $\mathbf{b}^*$ *not have a common neighbourhood curve.*

(i) If $\mathcal{P}_0(\mathbf{b}) \cap \mathcal{P}_0(\mathbf{b}^*) = \emptyset$*, then*

$$\mathcal{P}_s(\mathbf{b}) \cap \mathcal{P}_s(\mathbf{b}^*) = \begin{cases} \emptyset & \text{for } s < 1, \\ \partial\mathcal{P}_1(\mathbf{b}) \cap \partial\mathcal{P}_1(\mathbf{b}^*) & \text{for } s = 1. \end{cases}$$

(ii) If $\mathcal{P}_0(\mathbf{b}) \cap \mathcal{P}_0(\mathbf{b}^*) \neq \emptyset$*, then* $\mathcal{P}_0(\mathbf{b}) \cap \mathcal{P}_0(\mathbf{b}^*) = \{\mathbf{r}\}$ *where* $\mathbf{r}$ *is a common endpoint, and*

$$\mathcal{P}_s(\mathbf{b}) \cap \mathcal{P}_s(\mathbf{b}^*) = \begin{cases} \{\mathcal{D}_s(\mathbf{r})\} & \text{for } s < 1, \\ \partial\mathcal{P}_1(\mathbf{b}) \cap \partial\mathcal{P}_1(\mathbf{b}^*) \supseteq \{\mathcal{D}_1(\mathbf{r})\} & \text{for } s = 1. \end{cases}$$

PROOF. By construction, $\mathrm{conv}\{\mathbf{b}_i\}_i$ and $\mathrm{conv}\{\mathbf{b}_i^*\}_i$ intersect at most in common endpoints. If they had two endpoints in common, then, by convexity, also the straight-line segment joining them – a contradiction. This shows the initial claim in (ii). For the remaining claims, we defer the technicalities to the appendix and just outline the argument: By continuity of $s \mapsto \mathcal{D}_s$, any intersection point excluded by the statement of the proposition for $s \leq 1$ requires a contact of one ECC polygon's boundary with an ECC vertex of the other fragment at $s_1 < 1$ in a way that contradicts Lemma 11(i). □

COROLLARY 13. *Consider unrounded fragments* $\mathbf{b}$ *and* $\mathbf{b}^*$ *as above and their roundings* $\tilde{\mathbf{b}}$ *and* $\tilde{\mathbf{b}}^*$*.*

(i) If $\tilde{\mathbf{b}}_i = \tilde{\mathbf{b}}_j^*$*, then* $\mathbf{b}_i$*,* $\mathbf{b}_j^*$ *are unmerged control points or have already been equal before rounding.*

(ii) The set $H := \partial\mathcal{P}_1(\mathbf{b}) \cap \partial\mathcal{P}_1(\mathbf{b}^*)$ *is empty, a single point* $\{\tilde{\mathbf{p}}\}$*, or a line segment* $\overline{\tilde{\mathbf{p}}\tilde{\mathbf{q}}}$*.*

(iii) If $H = \{\tilde{\mathbf{p}}\}$*, then* $\tilde{\mathbf{p}}$ *is a control point of* $\tilde{\mathbf{b}}$ *and* $\tilde{\mathbf{b}}^*$*.*

(iv) If $H = \overline{\tilde{\mathbf{p}}\tilde{\mathbf{q}}}$*, then* $\tilde{\mathbf{p}}$ *and* $\tilde{\mathbf{q}}$ *are control points of* $\tilde{\mathbf{b}}$ *and* $\tilde{\mathbf{b}}^*$*, but no control point lies between them on* $\overline{\tilde{\mathbf{p}}\tilde{\mathbf{q}}}$*.*

(v) Any CC edge of $\tilde{\mathbf{b}}$ *or* $\tilde{\mathbf{b}}^*$ *that intersects* H *in more than one point is equal to* H*.*

PROOF. Claim (i) is immediate from Lemma 11(ii), noting that merged control points are external ECC vertices. Claim (ii) holds, because H is bounded, closed, convex, and of dimension at most 1. Claims (iii) and (iv) are direct consequences of Lemma 11(ii) in conjunction with Proposition 12. To see (v), observe that the intersection is again a line segment; by (iv), its endpoints cannot lie between the endpoints of H. □

REMARK 14. *The statements of the preceding proposition and corollary include the statements of Proposition 10 as special cases and are thus valid for any pair of fragments.*

We are now ready to prove that the rounded vertices and edges of $\mathcal{A}'$ really form an arrangement.

PROPOSITION 15. *Consider two meta-fragments* $\mathbf{c}$*,* $\mathbf{c}^*$ *and the edges of* $\mathcal{A}'$ *resulting from them after rounding. If the point sets* $\rho\mathbf{c}([0, 1])$ *and* $\rho\mathbf{c}^*([0, 1])$ *are not the equal, then* $\rho\mathbf{c}([0, 1]) \cap \rho\mathbf{c}^*([0, 1]) \subseteq \{\rho\mathbf{c}(0), \rho\mathbf{c}(1)\}$ *("intersections are endpoints").*

PROOF. If $\mathbf{c}$ is non-singleton, it is immediate from Corollary 13 that the polygonal enclosure of $\rho\mathbf{c}$ meets the enclosing polygon of any fragment of $\rho\mathbf{c}^*$ at most in $\rho\mathbf{c}(0)$ or $\rho\mathbf{c}(1)$. Hence let $\mathbf{c}$ be a singleton meta-fragment. If $\rho\mathbf{c}$ is not straight, $\rho\mathbf{c}((0, 1)) \subseteq \mathrm{int}\,\mathrm{conv}\{\rho(\mathbf{c})_i\}_i$ by Lemma 4(ii), and that set is disjoint from the polygonal enclosure of $\rho\mathbf{c}^*$ by Proposition 12. So let $\rho\mathbf{c}$ be straight and singleton. Lemma 4(i) states that $\rho\mathbf{c}([0, 1]) = \overline{\rho(\mathbf{c}_0)\,\rho(\mathbf{c}_n)}$. Consider any fragment $\tilde{\mathbf{b}}$ of $\rho\mathbf{c}^*$. If $\rho\mathbf{c}([0, 1])$ intersects the enclosing polygon of $\tilde{\mathbf{b}}$ other than in an endpoint, Corollary 13(v) implies that $\rho\mathbf{c}([0, 1])$ coincides with a CC edge $\overline{\tilde{\mathbf{b}}_{j_1}\,\tilde{\mathbf{b}}_{j_2}}$. In that case, we distinguish further whether $\tilde{\mathbf{b}}$ is straight. If not, $\overline{\tilde{\mathbf{b}}_{j_1}\,\tilde{\mathbf{b}}_{j_2}}$ does not contain a point of $\rho\mathbf{c}^*$ except maybe at its endpoints, so the claim holds. But if $\tilde{\mathbf{b}}$ is straight, then both $\rho\mathbf{c}$ and $\tilde{\mathbf{b}}$ have control polygons of the form

$$\rho(\mathbf{c}(0)), \ldots, \rho(\mathbf{c}(0)), \rho(\mathbf{c}(1)), \ldots, \rho(\mathbf{c}(1))$$

and hence are equal (as point sets). □

COROLLARY 16. *If* $\mathbf{c} \neq \mathbf{c}^*$ *and* $\rho\mathbf{c}([0, 1]) = \rho\mathbf{c}^*([0, 1])$*, then* $\rho\mathbf{c}$ *and* $\rho\mathbf{c}^*$ *are straight singleton meta-fragments whose control polygons consist of repetitions of one endpoint followed by repetitions of the other endpoint.*

PROPOSITION 17. *Let* $\rho\mathbf{c}$ *be a rounded meta-fragment. If* $0 < t < 1$*, then* $\rho\mathbf{c}'(t) \neq 0$ *("no interior irregular points"). If* $0 \leq t_1 < t_2 \leq 1$ *and* $\rho\mathbf{c}(t_1) = \rho\mathbf{c}(t_2)$*, then* $t_1 = 0$ *and* $t_2 = 1$ *("no interior self-intersection").*

PROOF. If $\mathbf{c}$ is an unrounded singleton meta-fragment consisting of one fragment $\mathbf{b}^1$, the claims are immediate from Lemma 3, since it was explicitly checked during conflict removal that $\rho\mathbf{c} = \rho\mathbf{b}^1$ is weakly bcp-monotone. If $\mathbf{c}$ consists of $\mathbf{b}^1, \ldots, \mathbf{b}^k$, $k > 1$, then each rounded fragment $\tilde{\mathbf{b}}^j$, being a subcurve of some neighbourhood curve, is bcp-monotone and thus regular and free of self-intersections at all points. It remains to argue that there is no interior self-intersection of $\rho\mathbf{c}$ arising from an intersection between two different rounded fragments $\tilde{\mathbf{b}}^{j_1}$, $\tilde{\mathbf{b}}^{j_2}$. If $\rho\mathbf{c}$ is straight, this is immediate, so we may assume now that $\rho\mathbf{c}$ is not straight. It folllows that $\tilde{\mathbf{b}}^{j_1}$, $\tilde{\mathbf{b}}^{j_2}$ are not straight and thus, with the exception of endpoints, contained in the disjoint interiors of their enclosing polygons by Lemma 4(ii). But Corollary 13(i) excludes collapse of merged fragment endpoints. □

The topology of an arrangement is given by the cyclic order of edges around vertices. So we show that the orientation of any three edges adjacent to one vertex in $\mathcal{A}'$, if not destroyed by collape of edges, agrees with the situation on the boundary of the hot pixel before rounding. In an arrangement of curves, there may be loops. Hence we do not consider entire edges $\rho\mathbf{c}$, consisting of fragments $\tilde{\mathbf{b}}^1, \ldots, \tilde{\mathbf{b}}^k$, but instead the *terminal fragment* $\tilde{\mathbf{b}}^1$ or $\tilde{\mathbf{b}}^k$ of each edge.

THEOREM 18. *Consider a vertex $\mathbf{a} \in \mathbb{Z}^2$ of $\mathcal{A}'$. Let $\rho\mathbf{c}^1$, $\rho\mathbf{c}^2$, $\rho\mathbf{c}^3$ be three rounded meta-fragments incident to $\mathbf{a}$. For $j = 1, 2, 3$, let $\tilde{\mathbf{b}}^j$ be a terminal fragment of $\rho\mathbf{c}^j$ incident to $\mathbf{a}$ so that the cyclic order of $(\tilde{\mathbf{b}}^1, \tilde{\mathbf{b}}^2, \tilde{\mathbf{b}}^3)$ around $\mathbf{a}$ is positive (i.e., counterclockwise). For $j = 1, 2, 3$, there exist intersection points $\mathbf{p}^j$ of the corresponding unrounded fragments $\mathbf{b}^j([0,1])$ and the pixel boundary $A := \partial(\mathbf{a}+U)$. Furthermore, any triple $(\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)$ of such points has positive cyclic order on the topological circle A.*

PROOF. Let $j \in \{1, 2, 3\}$. Exactly one of $\mathbf{b}^j(0)$, $\mathbf{b}^j(1)$ lies in $V = \mathbf{a}+U$: if $\mathbf{b}^j$ is merged, this follows from Lemma 9(ii); if $\mathbf{b}^j$ is unmerged, we know $\tilde{\mathbf{b}}^j$ is weakly bcp-monotone, so $\rho(\mathbf{b}_0^j) = \tilde{\mathbf{b}}^j(0) \neq \tilde{\mathbf{b}}^j(1) = \rho(\mathbf{b}_n^j)$, hence only one endpoint them rounds to $\mathbf{a}$, the other one lies outside V. In either case, $\mathbf{b}^j$ intersects $A = \partial V$ in some point $\mathbf{p}^j$. By Proposition 5, any choice of $\mathbf{p}^j \in \operatorname{conv}\{\mathbf{b}_i^j\}_i \cap A$ results in the same cyclic order of $(\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3)$, so the sets $\operatorname{conv}\{\mathbf{b}_i^j\}_i \cap A$ have a well-defined cyclic order on A; in particular, this is the cyclic order of any triple $\mathbf{p}^j \in \mathbf{b}^j([0,1]) \cap A$, $j = 1, 2, 3$. It remains to argue that this cyclic order of ECC polygons is preserved during rounding. This is because $\mathcal{D}_s$ shrinks pixel boundaries continuously for all $s \in [0, 1]$ and bijectively for $s < 1$. If no collapse happens at $s = 1$, the cyclic order is preserved. □

5. CONCLUDING REMARKS

We have presented a new algorithm for approximate arrangement computation of Bézier curves. Its salient and novel feature are the topological guarantees on the output; in particular, inversions of topology are avoided. Thereby, the algorithm produces a geometric rounding of the true arrangement induced by its input. To achieve this, we have replaced the ε-approximation of points in conventional intersection algorithms with locating points in a pixel grid, followed by generalized snap rounding of control polygons onto the grid. Regarding implementation, we remark that for input control points whose coordinates are integers, or more generally finite binary fractions, all control point coordinates constructed are also finite binary fractions; costly rational and algebraic coordinates are avoided.

Clearly, the number of subdivisions determines the complexity of the algorithm. Unfortunately, this number has no bound in terms of the number of input curves: even for two curves, any fixed number of subdivisions may be insufficient. Despite this shortcoming, intersection computation by repeated subdivision is very widely used in practice, and adding a guard against inversion of topology has its merits. It remains for future work to identify good parameters for a rigorous analysis.

We have given a deformation proof for topology preservation when rounding chains of fragments (Section 4), which is applicable to a wider class of algorithms. Such algorithms might drive subdivision and merging in more sophisticated ways, e.g., to allow a unique static definition of the output in terms of the input geometry, or to attain guarantees on the success of merging in terms of the input geometry. We expect that our proof technique may serve as a starting point for research into rounding arrangements of other classes of curves that are defined in terms of control points, or consist of Bézier pieces, such as B-splines.

Acknowledgement

We thank Kurt Mehlhorn for useful comments.

6. APPENDIX: PROOF DETAILS

PROOF OF **Proposition 12**. We already saw that $\mathcal{P}_0(\mathbf{b})$ and $\mathcal{P}_0(\mathbf{b}^*)$ intersect at most a in common endpoint but did not carry out the argument "by continuity". We begin with claim (i). Let N denote the total number of internal, boundary, and external vertices in the ECCs of $\mathbf{b}$ and $\mathbf{b}^*$. All their possible positions in the plane are given by N pairs of coordinates. Concatenating them, we get one vector in $\mathbb{R}^{2N}$. Vice versa, any vector in $\mathbb{R}^{2N}$ gives positions for the ECC vertices and thus defines two ECC polygons $\mathcal{Q}$ and $\mathcal{Q}^*$. We consider three subsets of $\mathbb{R}^{2N}$ defined by conditions on $\mathcal{Q}$ and $\mathcal{Q}^*$ as resulting from the respective vertex positions:

$$
\begin{aligned}
U_1 &= \{\emptyset = \mathcal{Q} \cap \mathcal{Q}^*\} && \text{("no intersection"),}\\
C &= \{\emptyset \neq \mathcal{Q} \cap \mathcal{Q}^* = \partial\mathcal{Q} \cap \partial\mathcal{Q}^*\} && \text{("boundary intersn."),}\\
U_2 &= \{\emptyset \neq \mathcal{Q} \cap \mathcal{Q}^* \neq \partial\mathcal{Q} \cap \partial\mathcal{Q}^*\} && \text{("interior intersn.").}
\end{aligned}
$$

Obviously, the sets U_1, C and U_2 form a partition of $\mathbb{R}^{2N}$. The sets U_1 and U_2 are open, because their defining conditions remain valid under sufficiently small perturbations of vertex locations. The set $C \cup U_2$ is the complement of U_1 and thus closed. Recall from elementary point-set topology that the union $U_1 \cup U_2$ of two disjoint non-empty open sets is *not connected*; in particular, there is no continuous path in $U_1 \cup U_2$ that connects a point in U_1 with a point in U_2; rather, any such connecting path has to pass through C.

The deformation $\mathcal{D}_s$, $s \in [0, 1]$, induces a continuous path $\delta\colon [0,1] \to \mathbb{R}^{2N}$ that maps s to the vector of ECC vertex coordinates of $\mathbf{b}$ and $\mathbf{b}^*$ at time s. By construction, $\delta(0) \in U_1$. We shall demonstrate that $\delta(s) \notin C \cup U_2$ for all $s < 1$.

Assume to the contrary that $\delta(s_1) \in C \cup U_2$ for some $s_1 < 1$. By compactness of the preimage $\delta^{-1}(C \cup U_2)$, we may choose s_1 to be minimal. We have $s_1 > 0$, because $\delta(0) \notin C \cup U_2$. As $\delta([0, s_1)) \subseteq U_1$, we have $\delta(s_1) \in C$. Hence $\emptyset \neq \partial\mathcal{P}_{s_1}(\mathbf{b}) \cap \partial\mathcal{P}_{s_1}(\mathbf{b}^*)$, so there is a pair of intersecting ECC edges $\overline{\mathbf{pq}}$ and $\overline{\mathbf{vw}}$ of $\mathbf{b}$ and $\mathbf{b}^*$, respectively. They do not intersect transversally, because a transversal intersection is preserved by sufficiently small perturbations of endpoint coordinates, which would entail $\delta(s_1 - \varepsilon) \notin U_1$ for small $\varepsilon > 0$, contradicting the minimality of s_1. Hence $\overline{\mathbf{pq}}$ and $\overline{\mathbf{vw}}$ intersect in an endpoint only or overlap; in either case, there is an endpoint of one lying on the other. This contradicts Lemma 11(i). We conclude that $\delta([0,1)) \subseteq U_1$ and $\delta(1) \in U_1 \cup C$ and have thus proved (i).

The argument for (ii) is similar but needs to take the following aspects into account. When counting the number N of ECC vertices, the common vertex $\mathbf{r}$ is counted only once. In all conditions on ECC polygons, "[no] intersection" becomes "[no] intersection other than $\mathbf{r}$". The disconnectedness argument for $s < 1$ is done in the topological space

$\mathbb{R}^{2N} \setminus E$ from which we have excluded the exceptional positions

$$E = \{\text{another ECC vertex coincides with } \mathbf{r}\}.$$

This is necessary, otherwise U_1 would not be open. (If another ECC vertex of $\mathcal{Q}$ is at point $\mathbf{r}$, an arbitrarily small movement may bring it into the interior of $\mathcal{Q}^*$.) Lemma 11(i) guarantees that $\mathcal{D}_s$ does not reach E for $s < 1$. □

7. REFERENCES

[1] M. de Berg, D. Halperin, and M. Overmars. An intersection-sensitive algorithm for snap rounding. *Comput. Geom.*, 36(4):159–165, 2007.

[2] O. Devillers and P. Guigue. Inner and outer rounding of set operations on lattice polygonal regions. In *Proc. 20th Annu. Sympos. Comput. Geom. (SCG'04)*, pages 429–437, 2004.

[3] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. On the design of CGAL, a computational geometry algorithms library. *Software – Practice and Experience*, 30:1167–1202, 2000.

[4] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, 4th edition, 1997.

[5] M. Goodrich, L. J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *Proc. 13th Annu. Sympos. Comput. Geom. (SCG'97)*, pages 284–293, 1997.

[6] D. H. Greene. Integer line segment intersection. Unpublished manuscript, cited after [8].

[7] D. H. Greene and F. F. Yao. Finite-resolution computational geometry. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS'86)*, pages 143–152, 1986.

[8] L. J. Guibas and D. H. Marimont. Rounding arrangements dynamically. *Internat. J. of Comput. Geometry & Applications*, 8:157–176, 1998.

[9] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. *Internat. J. of Comput. Geometry & Applications*, 14(4-5):277–310, 2004.

[10] D. Halperin and E. Packer. Iterated snap rounding. *Comput. Geom.*, 23(2):209–225, 2002.

[11] D. Halperin and C. R. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. *Comput. Geom.*, 10(4):273–287, 1998.

[12] J. Hershberger. Improved output-sensitive snap rounding. In *Proc. 22nd Annu. Sympos. Comput. Geom. (SCG'06)*, pages 357–366, 2006.

[13] J. D. Hobby. Practical segment intersection with finite precision output. *Comput. Geom.*, 13:199–214, 1999.

[14] L. Kettner and S. Näher. Two computational geometry libraries: LEDA and CGAL. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 65, pages 1435–1463. CRC Press LLC, Boca Raton, FL, 2nd edition, 2004.

[15] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[16] K. Mehlhorn, R. Osbild, and M. Sagraloff. Reliable and efficient computational geometry via controlled perturbation. In *Automata, Languages and Programming, 33rd Internat. Colloquium (ICALP'06), Part I*, volume 4051 of *LNCS*, pages 299–310. Springer, 2006.

[17] V. Milenkovic and E. Sacks. An approximate arrangement algorithm for semi-algebraic curves. In *Proc. 22nd Annu. Sympos. Comput. Geom. (SCG'06)*, pages 237–246, 2006. Full version to appear in *Internat. J. of Comput. Geometry & Applications*.

[18] V. J. Milenkovic. Shortest path geometric rounding. *Algorithmica*, 27:57–86, 2000.

[19] V. J. Milenkovic and L. R. Nackman. Finding compact coordinate representations for polygons and polyhedra. *IBM J. Res. Develop.*, 34(5):753–769, 1990.

[20] E. Packer. Iterated snap rounding with bounded drift. In *Proc. 22nd Annu. Sympos. Comput. Geom. (SCG'06)*, pages 367–376, 2006.

[21] H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and B-Spline Techniques*. Springer, 2002.

[22] L. Ramshaw. Blossoms are polar forms. Research Report 34, Digital, Systems Research Center, Palo Alto, CA, USA, January 1989. `ftp://ftp.digital.com/pub/DEC/SRC/research-reports/SRC-034.pdf`.

[23] C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 41, pages 927–952. CRC Press, Boca Raton, FL, 2nd edition, 2004.

[24] C. K. Yap. Complete subdivision algorithms, I: Intersection of Bezier curves. In *Proc. 22nd Annu. Sympos. Comput. Geom. (SCG'06)*, pages 217–226, 2006.