Conic Approximation of Convolution Curve

Tae-wan Kim
Seoul National University
Computer Aided Design and Analysis Laboratory
Seoul, 151-742 Korea
tkim@cad.snu.ac.kr

In-Kwon Lee
POSTECH
Information Research Laboratories
Pohang, 790-784 Korea
iklee@postech.ac.kr

Abstract

Given two planar curves, a convolution curve is computed by applying vector sums to all pairs of curve points which have the same curve normal direction. The convolution curve can be used to compute Minkowski sum of two planar objects which is important in various geometric computations such as collision detection and font design. In this paper, we present an algorithm to generate a conic approximation of convolution curve computed from two conics. Conics play an important role in geometric design to represent primitive objects such as parabola, ellipse, and hyperbola. Our algorithm can be directly applied to compute the boundary of a Minkowski sum of two planar objects bounded by piecewise conics. We derive the necessary condition of cusp and describe the method of cusp detection. As an application, we demonstrate the piecewise conic approximation of the boundary of a Minkowski sum computed from two input curves.

Keywords: convolution, conic, Minkowski sum, offset, Bézier curve, B-spline curve, rational quadratic curve

1. Introduction

Minkowski sum denoted by $P \oplus Q$ of two planar objects $P$ and $Q$ is the set of the vector sums generated by all pairs of points in two objects. Minkowski sum has been used to compute collision-free paths in robot motion planning [2, 10, 13, 15, 16]. In some areas such as metamorphosis [12] and font design [9], Minkowski sum is used as a modeling tool, because Minkowski sum generates a natural mixing of the shapes of two objects.

In spite of the simple definition, to compute the boundary of a Minkowski sum object is tricky especially when two given objects are bounded by curves. Let $A = \partial P$ and $B = \partial Q$ be boundary curves of two planar objects $P$ and $Q$, respectively. Lee, Kim, and Elber [15] used a fact that the problem of computing the boundary of a Minkowski sum $\partial(P \oplus Q)$ can be transformed into the problem of computing convolution curve $A * B$ [2]. The convolution curve $A * B$ is a trace of the vector sums generated from all pairs of curve points (one from $A$ and one from $B$) that have the same curve normal direction. When both $P$ and $Q$ are convex objects, $A * B$ is exactly the same as $\partial(P \oplus Q)$. However, $\partial(P \oplus Q)$ is a subset of $A * B$, in general [2] (see Figure 1). Lee et. al [15] eliminate the redundant parts of $A * B$ which do not contribute to $\partial(P \oplus Q)$. An important step in computation of $\partial(P \oplus Q)$ is to subdivide $A$ and $B$ into $A_i$, $i = 1, \cdots, n$, and $B_j$, $j = 1, \cdots, m$, respectively, such that each segment $A_i$ and $B_j$ participates at least one compatible pair $(A_k, B_l)$, such that $A_k$ and $B_l$ share exactly the same Gaussian image. This subdivision is called compatible subdivision [15]. The exact convolution curve of two compatible curve segments cannot be a rational curve so that various approximation methods are developed. Lee et. al [15] suggested and compared four types of approaches: control point based, interpolation based, polynomial quadratic curve approximation based, and reparametrization based.

In this paper, we focus on the problem to compute a conic approximation of the convolution curve generated from two compatible conics. Conics, also called conic sections, have been widely used in CAD/CAM industry and computer graphics to represent the primitive curve objects [7]. Aircraft industry and font design are the main application areas of conics; thus, it is very useful to compute the conic approximation of the boundary of a Minkowski sum of two planar objects bounded by piecewise conics. Conics can be classified as parabola, ellipse, and hyperbola. The parametric form of conics is a rational quadratic Bézier curve (see [14] for the details of the relation between conics and rational quadratic Bézier curves). A conic section has three control points so that the positions of these control points of the approximated convolution curve are fixed for given two input conics (see Section 3.1). A conic section has three more design parameters: three weights corresponding to the control points. Nevertheless, if a conic section is
nondegenerate, the weights of two end control points can always be transformed to 1 by standardization process; thus, we can assign 1 to the weights of two end control points. Our approach is a generalization of the conic offset approximation suggested by Farin [6]. Farin [6] determined the middle weight by exactly offsetting at parameter \( t = \frac{1}{2} \), for conic offset approximation. In this paper, we use the implicit form of conics to determine the middle weight of the conic approximation. The paper is organized as follows. In Section 2, we describe the basic definitions and properties of Minkowski sum, convolution curve, and conic section. Section 3 describes an algorithm to construct a conic approximation of the convolution curve of two conics including a method to estimate the approximation error. In Section 4, we show some examples and results of conic approximation of convolution curve as well as a demonstration of conic approximation of the boundary of a Minkowski sum computed from two ellipses. In Section 5 we discuss the convolution of convex and concave input pair conics. In Section 6 we show some examples of Minkowski sums for convex and concave objects. Finally, in Section 7, we conclude the paper and suggest further research problems.

2. Preliminaries

Let \( P \) and \( Q \) be planar curved objects respectively. Minkowski sum of \( P \) and \( Q \) is the set of the sum of all points in the two objects:

**Definition 1** The Minkowski sum is defined by

\[
P \oplus Q = \{ a + b \mid a \in P, \ b \in Q \}.
\]

Let \( A(t) \) and \( B(s) \) be regular planar parametric curves. A convolution curve \((A * B)(t)\) of \( A(t) \) and \( B(s) \) is defined in Definition 2. A parallel condition in the definition implies a convolution curve cannot be represented by a rational curve in general.

**Definition 2** The convolution \( A * B \) is defined by

\[
(A * B)(t) = A(t) + B\left(s\left(\frac{t}{2}\right)\right),
\]

where \( A'(t) \parallel B'(s(t)) \), and \( (A'(t), B'(s(t))) > 0 \), for a reparametrization \( s = s(t) \).

Let \( A \) and \( B \) represent the boundary curves of planar curved objects \( P \) and \( Q \), respectively. The relation between the convolution \( A * B \) and the boundary of the Minkowski sum \( \partial (P \oplus Q) \) is as follows [2]:

\[
\partial (P \oplus Q) \subset (A \oplus B) \subset (A * B).
\]

**Definition 3** A rational quadratic Bézier curve segment [7, 11, 17] is defined by

\[
C(t) = \frac{w_0 B_0^n(t) C_0 + w_1 B_1^n(t) C_1 + w_2 B_2^n(t) C_2}{w_0 B_0^n(t) + w_1 B_1^n(t) + w_2 B_2^n(t)},
\]

where \( 0 \leq t \leq 1 \),

\[
B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}
\]

are Bernstein polynomials. \( w_i \) are called weights, and \( C_i \) form the control polygon of \( C(t) \).

By utilizing Möbius parameter transformation, a rational quadratic Bézier curve can be reparameterized [7] by changing the weights according to

\[
\hat{w}_i = c^i w_i, \quad i = 0, \cdots, 2,
\]

where \( c \) is a nonzero constant. By using

\[
c = \sqrt{\frac{w_0}{w_2}}
\]

Figure 1. The boundary of a Minkowski sum and convolution curve. Left: two planar objects \( P \) and \( Q \), middle: \( A(=\partial P) * B(=\partial Q) \), right: \( \partial(\partial P \oplus Q) \)
and dividing each $\tilde{w}_i$ by $w_0$ we get the standard form of a rational Bézier curve which has unit weights at two end control points. Without loss of generality, we may assume that $w_0 = w_2 = 1$ for any nondegenerate conic section throughout the paper. We will also assume $w_1 > 0$.

**Definition 4** A standard conic section $C(t)$ (with $w_1 > 0$) is convex if the $z$-coordinate of the cross product in (6) is positive and concave if negative:

$$(C_1 - C_0) \wedge (C_2 - C_1),$$

where $C_i$ denotes control points of $A(t)$.

### 3. Conic Approximation of Convolution Curve

In this section, we consider two compatible conics $A(t)$, $0 \leq t \leq 1$, and $B(s)$, $0 \leq s \leq 1$, which share exactly the same Gaussian image [15]. When two curves are compatible, the normal vector field of one curve coincides with the normal vector field of the other. Assuming that the normal vectors of a conic section are pointing to the right hand side of the curve advancing direction, we define, for instance, the normal vector field of a curve $A(t) = (x(t), y(t))$, $0 \leq t \leq 1$, by

$$N_A(t) = \frac{(y'(t), -x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}} \in S^1,$$

where $N_A(t)$, a circular arc on the unit circle $S^1$, is called Gaussian image of $A(t)$. Thus, Gaussian images of two compatible conics are the same unit circular arc. Without loss of generality, we assume that $A'(0) \parallel B'(0)$ and $A'(1) \parallel B'(1)$.

The rest of this section is devoted to describe an algorithm which computes a conic approximation $(A \ast B)(t)$ of the convolution curve $A(t) \ast B(s)$. For simplicity of notation, let $C(t) = (A \ast B)(t)$. This convolution curve approximation algorithm is based on control point, which is an extension of the conic offset curve approximation method by Farin [6].

#### 3.1. Control Point Computation

The first step of the algorithm is to compute the control points of the approximated convolution $C(t)$. Since $A(t)$ and $B(s)$ are compatible, $A'(0) \parallel B'(0)$ and $A'(1) \parallel B'(1)$. Therefore, we can compute the exact convolution point at parameter 0 and 1 as follows:

$$C_0 = A_0 + B_0,$$

$$C_1 = A_1 + B_1,$$

$$C_2 = A_2 + B_2,$$

where $A_i$, $B_i$, and $C_i$, $i = 0, \cdots, 2$, denote the control points of $A(t)$, $B(s)$ and $C(t)$, respectively. The remaining control point $C_1$ of $C(t)$ is determined as the intersection of two tangent lines starting from $C_0$ and $C_2$. The direction of each tangent line is determined by $\overrightarrow{A_0A_1}$ and $\overrightarrow{A_2A_3}$ (or equivalently by $\overrightarrow{B_0B_1}$ and $\overrightarrow{B_2B_3}$), respectively. To show that $C_1 = A_1 + B_1$, we translate $B(s)$ so that $B_0$ moves on $A_0$. Figure 2 shows the conics after translation. For translated curves, let

$$B_1 = B_1 - (B_0 - A_0), \quad i = 0, \cdots, 2,$$

$$C_0 = C_0 - (B_0 - A_0),$$

$$C_2 = C_2 - (B_0 - A_0).$$

**Figure 2. Convolution after translation**

$\overrightarrow{B_0B_1}$ lies on the extension of $\overrightarrow{A_0A_1}$. Let $C_p$ be an intersection point of $\overrightarrow{C_2C_1}$ and a line emanating from $B_2$ which is parallel to $\overrightarrow{B_0B_1}$. Then, $\overrightarrow{A_0A_1} \parallel \overrightarrow{B_2C_p} \parallel \overrightarrow{A_0A_2} \parallel \overrightarrow{B_2C_2}$, and $\overrightarrow{A_2A_1} \parallel \overrightarrow{C_2C_0}$. $A_0$, $A_2$, $B_2$, and $C_2$ form a parallelogram since $\overrightarrow{C_2} = A_2 + B_2$. Besides, $\overrightarrow{A_0A_1} \parallel || \overrightarrow{B_2C_2} ||$, and $\overrightarrow{C_0C_2} \parallel || \overrightarrow{C_0C_2} ||$. Therefore,

$$\begin{align*}
\hat{C}_1 &= \hat{C}_2 + (\hat{C}_p - \hat{C}_2) + (\hat{C}_1 - \hat{C}_p) \\
&= \hat{C}_2 + (A_1 - A_2) + (B_1 - B_2) \\
&= A_2 + B_2 + (A_1 - A_2) + (B_1 - B_2) \\
&= A_1 + B_1.
\end{align*}$$

By translating back $\hat{C}_1$ to original position,

$$\begin{align*}
C_1 &= \hat{C}_1 + (B_0 - A_0) \\
&= A_1 + B_1 + (B_0 - A_0) \\
&= A_1 + B_2 - (B_0 - A_0) + (B_0 - A_0) \\
&= A_1 + B_1.
\end{align*}$$

#### 3.2. Tangent Parameter Computation

The middle weight of a conic section can be determined using implicit form of conics if three control points and
an arbitrary point on the conic section are given (see Sec-

Figure 3 shows the representation of an independent approach. This approach can be used for notes a cross product. However, we here try a coordinate in conic offset approximation suggested by Farin [6], an exact offset point \( O = A \left( \frac{1}{2} \right) + rN \left( \frac{1}{2} \right) \) is computed from the shoulder point \( A \left( \frac{1}{2} \right) \) of a given curve \( A(t), \) \( 0 \leq t \leq 1 \), where \( r \) and \( N(t) \) are an offset distance and a unit normal vector field of \( A(t) \), respectively. Farin [6] used the exact point \( O \) as a shoulder point of the offset approximation to compute a middle weight. In this paper, we select \( P_m = A(t_m) + B(s_m) \) as the extra point such that \( A'(t_m) \parallel B'(s_m) \parallel C_0C_2 \).

If both \( A(t) \) and \( B(s) \) lie on \( xy\)-plane and \( A'(t) \parallel C_0C_2 \), \( z\)-coordinate of \( A'(t) \wedge C_0C_2 \) vanishes, where \( (, \wedge ,) \) denotes a cross product. However, we here try a coordinate independent approach. This approach can be used for \( A(t) \) and \( B(s) \) both of which lie in three dimensional space.

For \( A(t) \), let \( T_0 = A_1 - A_0 \) and \( T_1 = A_2 - A_1 \). We can represent the given tangent vector \( C_0C_2 \) as a convex combination of vectors \( T_0 \) and \( T_1 \) as follows:

\[
\alpha A_0A_1 + \beta A_1A_2 = \alpha T_0 + \beta T_1.
\]

Figure 3 shows the representation of \( C_0C_2 \) as the convex combination of \( A_0A_1 \) and \( A_1A_2 \). If \( T_0 \) and \( T_1 \) are linearly independent and \( A_0, A_1, A_2, C_0, \) and \( C_2 \) are coplanar, we can compute unique \( \alpha \) and \( \beta \).

For \( A(t) \), let us denote

\[
A(t) = \frac{A_0B_0^2(t) + w_1A_1B_1^2(t) + A_2B_2^2(t)}{B_0^2(t) + w_1B_1^2(t) + B_2^2(t)} = \frac{a(t)}{w(t)}.
\]

The first derivative of a Bézier curve \( A(t) \) is represented as

\[
A'(t) = \frac{a(t)w'(t) - a(t)w(t)}{w(t)^2}.
\]

\[
= \frac{2}{w(t)^2} \left\{ (w_1t^2 - t^2 + t - 2w_1t + w_1)T_0 + (w_1t^2 - t^2 + t)T_1 \right\}.
\]

Note that numerator in Equation (14) has second degree only. The ratio of the coefficients of \( T_0 \) and \( T_1 \) is

\[
\begin{align*}
\frac{\alpha}{\beta} &= \frac{w_1t^2 - t^2 + 2w_1t + w_1}{w_1t^2 - t^2 + t}.
\end{align*}
\]

Equation (15) turns out a quadratic equation in \( t \). We take a root which is in \([0, 1] \) as \( t_m \). The tangent at the root outside \([0, 1] \) corresponds to a tangent of the complement segment of the conic section [7]. Using a similar technique, we can compute the parameter \( s_m \) of \( B(s) \).

3.3. Weight Computation

After we get an exact convolution point denoted by \( P_m = A(t_m) + B(s_m) \), we need to determine inner weight \( w_1 \) of \( C(t) \) where \( C(t) \) passes through the exact point.

The tangent of \( C(t) \) at \( P_m \) is not parallel to \( C_0C_2 \) in general although \( A'(t_m) \) and \( B'(s_m) \) are parallel to \( C_0C_2 \). We cannot use the four tangent theorem [7] of conics to determine \( w_1 \).

To determine \( w_1 \) we use the implicit form of conics. The exact convolution point \( P_m \) can be represented as \((\tau_0, \tau_1, \tau_2)\) where \( \tau_i \) is the barycentric coordinate of \( P_m \) with respect to a triangle \( \triangle C_0C_1C_2 \).

A conic section in implicit form [7] can be represented by

\[
\frac{\tau_1^2}{\tau_0^2} = \frac{\text{area}(P_m, C_0, C_2)^2}{\text{area}(P_m, C_0, C_2) \cdot \text{area}(P_m, C_1, C_0)} = 4(w_1^2).
\]

The equation (16) gives us immediately the positive inner weight \( w_1 \).

3.4. Error Computation

Once we get an approximation of convolution, we need to measure its error from the exact convolution. If an exact convolution \( P_i \) lies on \( C(t) \), \( C(t) \) has an exact convolution at the \( P_i \). However, in general, an exact convolution point \( P_i \) does not lie on the \( C(t) \) as long as exact convolution \((A \ast B)(t)\) cannot be represented by a conic section. To measure the error, we first sample some tangents from \( C(t) \) then measure the error for each sampled tangent. If maximal error exceeds a given tolerance, we subdivide the two input conics until the approximation meets the input tolerance. The following steps are necessary:
• The parameter domain \([0, 1]\) of \(C(t)\) is uniformly subdivided. Let \(m\) be the number of samples. We get \(m\) sample tangents \(C'(u_i); i = 0, \ldots, m - 1\), where \(u_i = \frac{i}{m-1}\). Let \(t_i\) and \(s_i\) be the parameters of \(A(t)\) and \(B(s)\) such that \(A'(t_i) \parallel C'(u_i)\) and \(B'(s_i) \parallel C'(u_i)\), respectively. \(P_i = A(t_i) + B(s_i)\) are \(m\) exact convolution points.

• One of the simplest error measures would be \(\|C(u_i) - P_i\|\), which represents the difference between an exact convolution point \(P_i\) and \(C(u_i)\) which has the same normal as \(P_i\), although the normal vector at \(C(u_i)\) may not pass through \(P_i\). Instead of this simple error estimation, we use the notion of normal error proposed by Degen [4]. Following Degen’s notation, we try to find a parameter \(v\) of \(C(t)\) such that \(C(v) = P_i + dN_i\), where \(N_i\) is a unit normal vector of \(C(t)\) at \(C(u_i)\) and \(-\infty < d < \infty\). Figure 4 shows the error computation of the approximated convolution.

To compute the \(d\) we utilize the implicit form of conics [7]. \(d\) represents the distance between approximation and exact convolution points. \(N_i\) indicates the direction from \(P_i\) to an approximated convolution. Let \(Q_i\) be \(P_i + dN_i\).

![Figure 4. Computing error of approximated conic convolution curve](image)

The subdivision occurs at the pair of points \(A(t_m)\) and \(B(s_m)\) on each iteration. \(A(t_m) + B(s_m)\) is a point on the exact convolution curve \((A + B)(t)\) because \(A'(t_m) \parallel \)

The parameter domain \([0, 1]\) of \(C(t)\) is uniformly subdivided. Let \(m\) be the number of samples. We get \(m\) sample tangents \(C'(u_i); i = 0, \ldots, m - 1\), where \(u_i = \frac{i}{m-1}\). Let \(t_i\) and \(s_i\) be the parameters of \(A(t)\) and \(B(s)\) such that \(A'(t_i) \parallel C'(u_i)\) and \(B'(s_i) \parallel C'(u_i)\), respectively. \(P_i = A(t_i) + B(s_i)\) are \(m\) exact convolution points.

One of the simplest error measures would be \(\|C(u_i) - P_i\|\), which represents the difference between an exact convolution point \(P_i\) and \(C(u_i)\) which has the same normal as \(P_i\), although the normal vector at \(C(u_i)\) may not pass through \(P_i\). Instead of this simple error estimation, we use the notion of normal error proposed by Degen [4]. Following Degen’s notation, we try to find a parameter \(v\) of \(C(t)\) such that \(C(v) = P_i + dN_i\), where \(N_i\) is a unit normal vector of \(C(t)\) at \(C(u_i)\) and \(-\infty < d < \infty\). Figure 4 shows the error computation of the approximated convolution.

To compute the \(d\) we utilize the implicit form of conics [7]. \(d\) represents the distance between approximation and exact convolution points. \(N_i\) indicates the direction from \(P_i\) to an approximated convolution. Let \(Q_i\) be \(P_i + dN_i\).

\[\tau_i^2 = \frac{\text{area}(Q_i, C_0, C_2)^2}{\text{area}(Q_i, C_2, C_1) \cdot \text{area}(Q_i, C_1, C_0)}\]
\[= \frac{\parallel (C_0 - Q_i) \land (C_2 - C_0) \parallel^2}{\parallel (C_2 - Q_i) \land (C_2 - C_1) \parallel} + \frac{1}{\parallel (C_1 - Q_i) \land (C_1 - C_0) \parallel}\]
\[= 4(w_1)^2. \quad (17)\]

Equation (17) produces a quartic equation in \(d\). All three cross products are parallel to the normal vector of \(\triangle C_0 C_1 C_2\). They are linearly dependent each other. Thus, equation (17) can be reduced to a quadratic form. Let \(M\) denote \((C_1 - C_0) \land (C_2 - C_1)\). To get a quadratic equation we compute \(\tau_i\) as follows:

\[\tau_1 M = 2(C_0 - P_i - dN_i) \land (C_2 - C_0),\]
\[\tau_0 M = 2(C_1 - P_i - dN_i) \land (C_2 - C_1),\]
\[\tau_2 M = 2(C_1 - P_i - dN_i) \land (C_1 - C_0). \quad (18)\]

Each \(\tau_i\) is a linear function of \(d\); for \(xy\)-planar curves \(z\) coordinates are taken. From equation (17) we get

\[\frac{\tau_i^2}{\tau_0 \tau_2} = 4(w_1)^2. \quad (19)\]

Equation (19) turns out a quadratic equation in \(d\). We select one of two roots depending on the direction of \(N_i\). We compute two points \(C(v) = P_i + dN_i\) on each root of \(d\) and measure the distance between \(P_i\) and \(C(v)\). A root \(d\) which generates a shorter distance is the right solution.

In previous sections, we computed the middle weight \(w_1\) under the condition that \(C(t)\) passes through the exact convolution point \(P_m = A(t_m) + B(s_m)\), where \(A'(t_m) \parallel B'(s_m) \parallel C_0 C_2\). Thus, the conic approximation \(C(t)\) has no error at the point \(P_m = C(t_*)\) such that \(C'(t_*) \parallel C_0 C_2\) (see Figure 6 and 8).

Suppose a given tolerance of approximation error is relatively big compared to the size of the bounding box of two given input curves \(A(t)\) and \(B(t)\). For instance, we can consider the case when the given tolerance is bigger than the sum of the size of the bounding boxes of two input curves. Then, we do not have to compute the above sampled approximation error, because the approximation \(C(t)\) always meets the given tolerance.

3.5. Algorithm

The algorithm to approximate the conic convolution curve is summarized in Algorithm 1. The algorithm recursively subdivides input conics until the approximation error is less than given tolerance \(\epsilon\). Note that we can take a subsegment from a given conic section with Eve information [7] without any error in the segmentation process. For the examples shown in Section 4, we implemented the recursive functions in Algorithm 1 using stack.

The convergency of the algorithm ConicConvolution can be proven by following theorem.

**Theorem 1** For two compatible conics, the piecewise approximation of conic convolution curve generated by Algorithm 1 converges to the exact convolution curve.

**Proof:** The subdivision occurs at the pair of points \(A(t_m)\) and \(B(s_m)\) on each iteration. \(A(t_m) + B(s_m)\) is a point on the exact convolution curve \((A + B)(t)\) because \(A'(t_m) \parallel \)
Algorithm 1

Input:
\[ A(t), 0 \leq t \leq 1, \text{and} \]
\[ B(s), 0 \leq s \leq 1; \text{two regular compatible standard conic curves;} \]
\[ \epsilon (> 0); \text{maximal error tolerance of approximation;} \]

Output:
\[ (A \ast^0 B)(t), 0 \leq t \leq 1; \text{conic approximation of the convolution curve of} \]
\[ A(t) \text{and} B(s); \]

Algorithm : ConicConvolution \((A(t), B(s), \epsilon)\)

\[
\begin{align*}
\text{begin} & \\
C_i & \leftarrow A_i + B_i \text{ for } i = 0, 1, 2, \text{where} \ A_i, B_i, \text{and} \ C_i \text{ are} \\
& \quad \text{control points of} \ A(t), B(s), \text{and} \ (A \ast^0 B)(t) \text{ respectively; } \\
w_0 & \leftarrow \text{weight of} \ C_0; \\
w_2 & \leftarrow \text{weight of} \ C_2; \\
t_m & \leftarrow \text{a parameter of} \ A(t) \text{ such that} \ A'(t) \parallel \overrightarrow{C_0C_2} \text{ and } 0 \leq t_m \leq 1; \\
s_m & \leftarrow \text{a parameter of} \ B(s) \text{ such that} \ B'(s) \parallel \overrightarrow{C_0C_2} \text{ and } 0 \leq s_m \leq 1; \\
C_m & \leftarrow A(t_m) + B(s_m); \\
r_0 & \leftarrow \text{side barycentric coordinate of} \ C_m \text{ on} \ \triangle C_0C_1C_2; \\
r_1 & \leftarrow \text{side barycentric coordinate of} \ C_m \text{ on} \ \triangle C_0C_1C_2; \\
r_2 & \leftarrow \text{side barycentric coordinate of} \ C_m \text{ on} \ \triangle C_0C_1C_2; \\
w_1 & \leftarrow \frac{2}{\sqrt{r_0r_2}}; \\
\epsilon_* & \leftarrow \text{maximal error between approximated} \ (A \ast^0 B)(t) \text{ and exact} \ (A \ast B)(t) \text{ by sampling;} \\
\text{if} \ \epsilon_* < \epsilon & \text{ then} \\
& \quad \text{return} \ (A \ast^0 B)(t); \\
\text{else} & \text{ begin} \\
& \quad \text{subdivide} \ A(t) \text{ at} \ t_m; \\
& \quad \text{subdivide} \ B(s) \text{ at} \ s_m; \\
& \quad \text{Reparameterize each} \ A^1(t), A^2(t), B^1(s), \text{and} \ B^2(s) \text{ to be in standard form;} \\
& \quad \text{return} \ 
\begin{align*}
\text{MergeConics (ConicConvolution} & (A^1(t), B^1(s), \epsilon), \\
& \quad \text{ConicConvolution} (A^2(t), B^2(s), \epsilon)); \\
\end{align*}
\text{end}
\end{align*}
\]

\[ B'(s_m). \text{ Thus, two end points of} \ C(t) \text{ are always on the} \]
\[ \text{exact convolution curve on each iteration. When the number} \]
\[ \text{of subdivision is approaching infinity, The piecewise} \]
\[ \text{approximation of conic convolution converges to the exact} \]
\[ \text{convolution curve.} \]

4. Examples of convolutions of convex conics

4.1. Conic Convolution Approximation

In this section, we show two examples using the proposed algorithm. In Figure 5, two input curves are \(\wedge\) shaped curves. The length of \(\overrightarrow{C_0C_2}\) is 15.0. Figure 6 shows an unsigned error function of the approximated convolution curve in Figure 5. The maximal error of the approximation is 0.023174. Note that the approximation error vanishes at the point \(P_m = A(t_m) + B(s_m)\) which is used as an extra exact convolution point to compute \(w_1\) of \((A \ast^0 B)(t)\). In the second example in Figure 7, we use two \(\wedge\) shaped input curves. The length of \(\overrightarrow{C_0C_2}\) is 15.03329638. The maximal error of the approximation in Figure 8 is 0.026088.
Figure 5. Conic convolution approximation of the first example

Figure 6. Error function of the first example

Figure 7. Conic convolution approximation of the second example

Figure 8. Error function of the second example
4.2. Minkowski sum of two ellipses

For an ellipse in general position on the plane, we can start with four piecewise conics where all control polygon legs form a rectangular box. Figure 9 shows an example of the approximation of the boundary of a Minkowski sum computed from two ellipses. One ellipse is a unit circle $x^2 + y^2 = 1$, and the other is $(x/5)^2 + (y/2)^2 = 1$ with $-20$ degree rotation. Tolerance of approximation error is $10^{-1}$.

Figure 10. The Same Inputs with Figure 9. Tolerance of approximation error is $10^{-3}$

Figure 11. Conic approximation for ellipses: $(x/2)^2 + y^2 = 1$ with 60 degree rotation and $(x/5)^2 + (y/4)^2 = 1$ with $-20$ degree rotation. Tolerance of approximation error is $10^{-4}$.

The curve approximation at the left-bottom and right-top corners has less subdivisions than others.

5. Convolution of Convex and Concave Conics

In the section 3, we described the convolution of two compatible convex conic objects. The orientation of the Gaussian image of a convex curve is counter clockwise, whereas that of a concave curve is clockwise. For a pair of convex-concave or concave-convex conic objects, the resulting convolution curve may have a cusp. In this section, we first study the necessary condition of the existence of a cusp on a convolution curve in terms of two given conic sections. We then give a formula to detect the parameter of cusp and the cusp point. We show several examples.

5.1. Curvature of Convolution curve

For a convex/concave or a concave/convex input conic pair, we realized that the convolution curve $(A \ast B)(t)$ has a cusp if the sign of the curvature of $(A \ast B)(t)$ at $t = 0$ is opposite to the sign of curvature at $t = 1$. In such case, we must split two input conics at parameters $t$ of $A(t)$ and $s$ of $B(s)$ where the cusp occurs. Given a parameter $t$ of $A(t)$, its compatible parameter $s$ of $B(s)$ is obtained from equation (26).
In this subsection, we derive the signed curvature of the exact convolution \((A + B)(t)\). While on the other, there is no cusp for a convex/convex or a concave/concave input conic pair. Let \(C(t) = (A + B)(t)\) for the simplicity of notation in deriving a cusp formula. In addition, let \(C(t) = (C_x(t), C_y(t))\). The curvature of a planar parametric curve \(C(t)\) is defined as [7]:

\[
\kappa(t) = \frac{C''_x(t)C'_y(t) - C'_x(t)C''_y(t)}{[C'_x(t)^2 + (C'_y(t))^2]^{3/2}}. \tag{20}
\]

First derivative of \(A(t)\) is:

\[
A'(t) = \frac{dA(t)}{dt} = 2a_2t^2 + a_1t + a_0, \tag{21}
\]

where,

\[
\begin{align*}
a_2 &= (w_1 - 1)(A_2 - A_0), \\
a_1 &= A_2 - 2w_1A_1 + (2w_1 - 1)A_0, \\
a_0 &= (A_1 - A_0)w_1.
\end{align*}
\]

Second derivative of \(A(t)\) is:

\[
A''(t) = \frac{d^2A(t)}{dt^2} = \frac{2}{w(t)^3}[(2a_2t + a_1)w(t) - 2(a_2t^2 + a_1t + a_0)]. \tag{22}
\]

Similarly for \(B(s)\),

\[
B'(s) = \frac{dB(s)}{ds} = 2b_2s^2 + b_1s + b_0, \tag{23}
\]

where,

\[
\begin{align*}
b_2 &= (w_1 - 1)(B_2 - B_0), \\
b_1 &= B_2 - 2w_1B_1 + (2w_1 - 1)B_0, \\
b_0 &= (B_1 - B_0)w_1.
\end{align*}
\]

\[
B''(s) = \frac{d^2B(s)}{ds^2} = \frac{2}{w(s)^3}[(2b_2s + b_1)w(s) - 2(b_2s^2 + b_1s + b_0)]. \tag{24}
\]

Let \(z\) be the coordinate of \(A'(t) \wedge B'(s)\) as follows:

\[
z(s, t) = a_x(t)b_y(s) - b_x(t)a_y(s)
\]

\[
= (a_2x^2 + a_1x + a_0x)(b_2y^2 + b_1y + b_0y)
\]

\[
- (b_2x^2 + b_1x + b_0x)(a_2y^2 + a_1y + a_0y)
\]

\[
= c_2(t)s^2 + c_1(t)s + c_0(t), \tag{25}
\]

where

\[
c_2(t) = (a_2x^2 + a_1x + a_0x)b_2y
\]

\[
- (a_2y^2 + a_1y + a_0y)b_2x
\]

\[
= (a_2x^2 - a_2y^2)b_2y + (a_1x^2 - a_1y^2)b_2x
\]

\[
+ (a_0x^2 - a_0y^2)b_2x,
\]

\[
c_1(t) = (a_2x^2 + a_1x + a_0x)b_1y
\]

\[
- (a_2y^2 + a_1y + a_0y)b_1x
\]

\[
= (a_2x^2b_2y - a_2y^2b_2x) + (a_1x^2b_1y - a_1y^2b_1x)t
\]

\[
+ (a_0x^2b_1y - a_0y^2b_1x)t
\]

\[
c_0(t) = (a_2x^2 + a_1x + a_0x)b_0y
\]

\[
- (a_2y^2 + a_1y + a_0y)b_0x
\]

\[
= (a_2x^2 - a_2y^2)b_0y + (a_1x^2 - a_1y^2)b_0x
\]

\[
+ (a_0x^2 - a_0y^2)b_0x.
\]

The derivatives are:

\[
\dot{c}_2(t) = 2(a_2x^2b_2y - a_2y^2b_2x)t + (a_1x^2b_1y - a_1y^2b_1x),
\]

\[
\ddot{c}_1(t) = 2(a_2x^2b_2y - a_2y^2b_2x) + (a_1x^2b_1y - a_1y^2b_1x),
\]

\[
\ddot{c}_0(t) = 2(a_2x^2b_2y - a_2y^2b_2x).
\]

The necessary condition of compatibility of input conics \(A(t)\) and \(B(s)\) is

\[
z(s, t) = c_2(t)s^2 + c_1(t)s + c_0(t) = 0, \tag{26}
\]

\[
s = -\frac{c_1(t)}{2c_2(t)} \pm \frac{\sqrt{c_1(t)^2 - 4c_2(t)c_0(t)}}{2c_2(t)},
\]

where, \(s \in [0, 1]\).

We can differentiate both sides in (26).

\[
\frac{dz(s, t)}{dt} = \dot{c}_2(t)s^2 + 2c_2(t)s\frac{ds}{dt} + \dot{c}_1(t)s + \dot{c}_0(t),
\]

\[
= \frac{ds}{dt}[2\dot{c}_2(t)s + c_1(t)] + \dot{c}_2(t)s^2 + \dot{c}_0(t) = 0. \tag{27}
\]

From (27),

\[
\frac{ds}{dt} = -\frac{\dot{c}_2(t)s^2 + \dot{c}_1(t)s + \dot{c}_0(t)}{2c_2(t)s + c_1(t)}. \tag{28}
\]

(28) is used in (31) and (32).
The second derivative of convolution curve is

\[ +4c_2(t)s \frac{ds}{dt} + 2c_1(t) \left( \frac{ds}{dt} \right)^3 + \hat{c}_1(t)s \frac{ds}{dt} + \hat{c}_0(t) = 0. \]  

(29)

\[
\frac{d^2s}{dt^2} = -\frac{1}{2c_2(t)s \frac{d^2s}{dt^2} + c_1(t)} (\hat{c}_2(t)s^2 + 4\hat{c}_2(t)s \frac{ds}{dt} + 2c_2(t) \left( \frac{ds}{dt} \right)^3 + \hat{c}_1(t)s \frac{ds}{dt} + \hat{c}_0(t)) .
\]  

(30)

Equations in (31) and (32) are used to compute cusp of the convolution curve \((A * B)(t)\).

5.2. Detection of a Cusp

We propose a simple form to check the condition of the existence of a cusp in \((A * B)(t)\). For standard conics \(A(t)\) and \(B(s)\) where each weight of inner control point \(A_i\) and \(B_i\) respectively is positive, if the following condition is true, \((A * B)(t)\) has a cusp in \(t \in [0, 1]\):

\[
(\langle A_1 - A_0 \rangle \wedge \langle A_2 - A_1 \rangle, (\langle A_1 + B_1 \rangle - (A_0 + B_0)) \\
\wedge (\langle A_2 + B_2 \rangle - (A_1 + B_1))) < 0,
\]  

(33)

where \(\langle , \rangle\) denotes a dot product. Once there is a cusp in \((A * B)(t)\) of two compatible conics \(A(t)\) and \(B(s)\), we must compute a cusp point and the corresponding parameters \(t\) and \(s\).

Let us consider a problem of minimization of a function \(f(t) = \| \frac{d(A * B)(t)}{dt} \| \|^2\). The necessary condition of a cusp of a convex-concave convolution \((A * B)(t)\) is \(f(t) = 0\). The derivative function is \(\frac{d}{dt}(A * B)(t) = 2 \left( \frac{d^2(A * B)(t)}{dt^2}, \frac{d(A * B)(t)}{dt} \right)\).

Equations (31) and (32) are used in \(f(t)\) and \(\frac{d}{dt}(A * B)(t)\). Acton [1] and Brent [3] discuss algorithms for minimization of one dimensional function. After ending \(t\) from \(f(t) = 0\), we compute the corresponding \(s\). We then subdivide both given \(A(t)\) and \(B(s)\) at \(t\) and \(s\), respectively.

5.3. Convolution of concave and convex conics without cusps

Given concave \(A(t)\) and convex \(B(s)\), the convolution does not have any cusps, where the signs of the curvature of \((A * B)(t)\) are same at both \(t = 0\) and \(t = 1\). Figure 12 shows an example of the convolution of concave and convex conics without any cusps, where (33) is not satisfied.

5.4. Convolution of concave and convex conics with a cusp

Given concave \(A(t)\) and convex \(B(s)\), the convolution has a cusp if it has opposite signed curvature value. Figure 13 shows an example of the convolution of concave and convex conics with a cusp, where (33) is satisfied.

Figure 12. Convolution of concave and convex objects

Figure 13. Convolution of concave and convex objects
5.5. Convolution of convex and concave conics with a cusp

Given convex \( A(t) \) and concave \( B(s) \), \((A \ast B)(t)\) has a cusp if the convolution curve has opposite signed curvature value each other. Figure 14 shows an example of the convolution of convex and concave conics with a cusp, where (33) is satisfied.

![Figure 14. Convolution of convex and concave objects](image)

6. Examples of the Minkowski sum with concave objects

In Figure 15, we give an example of the Minkowski sum of a convex and a concave objects bounded by piecewise conic segments, respectively. \( A(t) \) and \( B(s) \) are a convex and a concave (piecewise conic) objects, respectively. The (untrimmed) Minkowski sum \((A \ast a B)(t)\) is computed using the method described in [15]. First, the boundary curves of two given objects are subdivided and collected into a set of compatible conic pairs. Then, the conic convolution segment for each pair of compatible conics is computed using our method described in this paper. The untrimmed Minkowski sum can be trimmed by eliminating redundant parts to generate a trimmed boundary of a Minkowski sum such as shown in Figure 16. For the trimming, we also follow the same procedure presented in [15], which used the planesweep algorithm. Note that in the trimming step, each conic convolution approximation is tested by the existence condition of cusps. If any cusp exists, the convolution curve is more subdivided at the cusp.

![Figure 15. Minkowski sum of convex and concave objects](image)

![Figure 16. Convex and concave objects (Trimmed)](image)

7. Conclusion

This paper presented a method to approximate a convolution curve of given two conics. The output of this method is also a conic section. Since conics are important in geometric design, this algorithm can be used in various geometric applications.

As an application of the conic approximation of the convolution of two conics, we considered convex as well as concave conic objects to demonstrate the conic approximation of the boundary of a Minkowski sum. We also derived
In Section 3.2, we used an exact convolution point \( P_m = A(t_m) + B(s_m) \) such that \( A(t_m) \parallel B(s_m) \parallel C_0C_2 \) to compute the middle weight of the conic convolution curve. This method could be working good for the conics which have isosceles triangle as their control polygons. In the stage of selecting \( P_m \), we can reflect some shape information of conics to minimize the total error. For example, Farouki [8] studied on how to pick a parameter \( t \) for the optimal conic approximation of an offset curve using the given shape information of conics. The similar problem for the conic convolution approximation is being investigated.

8. Acknowledgements

The authors would like to thank Professor Myung-Soo Kim in Seoul National University of Korea for giving many pieces of advice. The authors acknowledge the financial support from National Research Laboratory grant from The Korean Ministry of Science and Technology, and the grant No. 2UD99520 from POSTECH Information Research Laboratories.

References


