Hardware Assistance for Constrained Circle Constructions I: Sequential Problems

Ching-Shoei Chiang,[1] Christoph M. Hoffmann[2] and Paul Rosen[3]

[1]Soochow University, chiang@scu.edu.tw
[2]Purdue University, cmh@cs.purdue.edu
[3]Purdue University, rosen@purdue.edu

ABSTRACT

In geometric constraint solving, constructing circles with indeterminate radius is an important sub problem. Such constructions are both *sequential*, meaning that we seek a circle tangent to three known geometric entities, as well as *simultaneous*, when several sets of entities, among them variable-radius circles, must be determined together.

In Part I, we investigate techniques to solve sequential construction problems of variable-radius circles, analyzing the case when at least one of the constraining entities is a Bézier curve. We consider first an algebraic solution in which we restrict to *Pythagorean hodographs* (PH). However, the polynomial degrees become very large, rendering this approach impractical. So, we develop an approach in which the needed computations are assisted by graphics hardware commonly available on PCs and laptops. Here we achieve greater generality, allowing arbitrary curves, greater numerical stability, and extreme speed-ups.

Keywords: geometric constraint solving, variable-radius circles, cyclographic maps, PH curves, equation solving, hardware acceleration, GPU.
DOI: 10.3722/cadaps.2010.17-32

1 INTRODUCTION

Geometric constraint solving plays a pivotal role in computer-aided design (CAD) as well as in dynamic geometry software. Practical solvers include graph-theoretic solvers in which the constraint problem is decomposed into sub problems of known structure, those sub problems subsequently are solved, and then the solution fragments are assembled into a solution of the original problem; e.g., [13],[19]. For 2D solvers in particular, the construction phase becomes demanding when one considers placing variable-radius circles and extends the geometric vocabulary to include entities such as PH curves; see [5],[9],[11]. For such constructions, the dominant approach is algebraic. This approach investigates the equation system that arises from the construction problem and, through algebraic and geometric transformations, changes it into an equivalent system that is triangular, so reducing solving the system to root finding. In many cases, but not in all, this reduction is capable of delivering low-degree polynomials that can be solved efficiently and simply; e.g. [1]. Where the systems become algebraically complicated, the problem of finding a practical way to solve them arises. In those cases, commercial solvers will either give up or else employ numerical approximations; e.g. [19]. Those computations may

not be stable unless special techniques are employed; e.g., [7],[17]. There is an extensive literature on geometric constraint solving, PH curves, and on Bezier curves. We refer the reader to [24] for a recent survey.

In this paper, we revisit the problem of constructing variable-radius circles where tangency is required to high-order entities. In Part 1, we concentrate on the sequential case. In [6], we derived an algebraic transformation that puts tangency to PH-curves into reach, considering combinations of PH curves with lines and known circles, and approaching the problem algebraically. In this paper, we derive first the algebra for simplifying the equation systems that arise and making them solvable in principle. However, as the algebraic degrees of such systems grow considerably, and since many of the solutions from such systems are unwanted on geometric grounds, we develop a purely geometric method for solving the equations that relies on graphics hardware assistance that is readily available in most personal computers and workstations, and even in many laptops. This hardware-assisted geometric approach has significant advantages laid out later in section 4, including an expansion of the domain of tractable problems, as well as performing at blinding speeds.

The remainder of the paper is structured as follows. In Section 2 we define the problems we consider algebraically and review prior results needed to accomplish our task. Section 3 develops the algebra for solving the constructions involving two or three PH-curves. Section 4 then addresses the problem how to practically solve the associated equation systems. Those solutions are based on using graphics-card hardware, allowing a highly parallel solution that is numerically stable. This section also considers generalizations that become possible because of this solution method.

In Part 2, we extend the scope of our techniques by considering cluster merging in which one of the clusters is a variable-radius circle. This problem was previously considered purely algebraically in [3],[14],[15] and leads to equation systems that, even after careful factoring, require solving univariate polynomials of relatively high degree.

2 DEFINITIONS AND THEOREMS

2.1 Planar PH Boundary

The *hodograph* of a parametric plane curve is the locus described by the first derivatives of the curve. A *Pythagorean Hodograph* (PH) curve is a parametric curve that has a polynomial hodograph. That is, if the curve is $r(t) = [x(t), y(t)]$, and its derivative is $r'(t) = [x'(t), y'(t)]$ then the length of the derivative $\sigma(t) = \sqrt{x'^2(t) + y'^2(t)}$ is polynomial. A PH has many properties, among them polynomial arc length, rational curvature and rational offset [10]. Moreover, the *cyclographic map* [5] of a PH has a polynomial form. From [5],[6] we recall the following:

Definition 1: (Complex representation) [9]

A PH curve in the complex plane is a complex valued polynomial curve $r(t)$ of the form $r(t) = x(t) + iy(t)$. There must exist a complex polynomial $\omega(t) = u(t) + iv(t)$, such that the hodograph is $r'(t) = \omega^2(t) = u^2(t) - v^2(t) + 2iu(t)v(t)$.

Although the hodograph so defined is in the complex plane, we can map the real and imaginary parts into the Cartesian x and y coordinates, respectively. Thus, the hodograph of the PH curve $r(t) = [x(t), y(t)]$ can be represented as $r'(t) = [u^2(t) - v^2(t), 2u(t)v(t)]$, so that the polynomial $\sigma(t)$ has the form $\sigma(t) = u^2(t) + v^2(t)$.

Theorem 1: Representing r'(t), x'(t), y' (t) in Bézier form [6]

Given two polynomials $u(t)$ and $v(t)$ of degree μ in Bernstein-Bézier form, where $u(t) = \sum_{k=0}^{\mu} B_k^{\mu}(t) u_i$ and $v(t) = \sum_{k=0}^{\mu} B_k^{\mu}(t) v_i$, we obtain the hodograph

$$r'(t) = [x'(t), y'(t)] = [u^2(t) - v^2(t), 2u(t)v(t)] = \left[\sum_{k=0}^{2\mu} B_k^{2\mu}(t) I_k, \sum_{k=0}^{2\mu} B_k^{2\mu}(t) J_k\right] \quad (1)$$

for a PH curve and its tangent length $\sigma(t) = \sqrt{x'^2(t) + y'^2(t)} = u^2(t) + v^2(t) = \sum_{k=0}^{2\mu} B_k^{2\mu}(t) H_k$; where I_k, J_k, and H_k are:

$$H_k = \sum_{i=0}^{k} \varphi_{i,k-i}^{\mu,\mu}(u_i u_{k-i} + v_i v_{k-i}), \quad k = 0...2\mu$$

$$I_k = \sum_{i=0}^{k} \varphi_{i,k-i}^{\mu,\mu}(u_i u_{k-i} - v_i v_{k-i}), \quad k = 0...2\mu$$

$$J_k = \sum_{i=0}^{k} \varphi_{i,k-i}^{\mu,\mu}(u_i v_{k-i} + v_i u_{k-i}), \quad k = 0...2\mu \tag{2}$$

$$\varphi_{i,j}^{m,n} = \begin{cases} 0 & if\ i < 0\ or\ j < 0 \\ \dfrac{C_i^m C_j^n}{C_{i+j}^{m+n}} & otherwise \end{cases}$$

Theorem 2: (PH curve in Bézier form)
For the hodograph $r'(t)$ in Theorem 1, the corresponding PH curve is a polynomial curve of degree 2μ+1. Given the choice of the point p_0, the PH curve in Bézier form is $B(t) = \sum_{i=0}^{2\mu+1} B_i^{2\mu+1}(t)p_i$, where the control points p_{k+1} are given by $p_{k+1} = p_k + 1/(2\mu+1)[I_k, J_k]$, for $k = 0...2\mu$.

The theorem follows directly from the derivative property of Bézier curves.

2.2 Cyclographic Maps

Cyclographic maps have been used as early as 1929; [18]. We are not using the full generality of that concept, however. For our purposes, the machinery developed in [14] suffices. As illustrative example, the Apollonius problem for three given circles is to find the 8 tangent circles to the three circles. When the three circles are oriented and using cyclographic maps, we can divide the Apollonius problem into 4 sub-problems, each determining two tangent circles [14]. Using a similar approach, we will solve the Apollonius problem for three geometric entities chosen from lines, circles, and PH curves. We consider lines, circles, and PH curves as rays (oriented lines), cycles (oriented circles) and oriented PH curves. We also call the cyclographic map of an entity L the *γ-map* of the entity, denoted $\gamma(L)$.

Definition 2: (The γ-maps for a Ray) [3]

An oriented line is called a *ray*. For a ray $L: Ax + By + D = 0$ in the *xy*-plane, the vector (A,B) is the normal vector on the left hand side of the ray, so the orientation of the ray is thereby assigned. The *γ-map of the ray L* is the plane $\gamma(L): Ax + By - \sqrt{A^2 + B^2}z + D = 0$ and is a plane in 3-space inclined 45 degrees against the *xy*-plane.

Definition 3: (The γ-maps for a Cycle) [3]

An oriented circle is called a *cycle*. For a cycle $c: (x-a)^2 + (y-b)^2 = r^2$ in the *xy*-plane, r is called the *oriented radius*. If $r > 0$, the cycle is counterclockwise, and clockwise for $r < 0$. The *γ-map of the cycle c* is the right-circular cone $\gamma(c): (x-a)^2 + (y-b)^2 - (z-r)^2 = 0$ in 3-space whose generators are inclined 45 degrees against the *xy*-plane.

Definition 4: (Rotated normal vector fields of PH boundary) [5]

For a 2D PH boundary $B(t) = [x(t), y(t), 0]$ in the *xy*-plane, we let the orientation be induced by the parameterization. The *normal* vector field on the left hand side is given by $[-y'(t), x'(t), 0]$. Because of $\sigma(t) = \sqrt{x'^2 + y'^2}$, the vector field $V(t) = [-y'(t), x'(t), \sigma(t)]$ is inclined 45 degrees against the *xy*-plane. We call $V(t)$ the *rotated normal vector field* of $B(t)$, or vector field of $B(t)$ in short.

Theorem 3: (Rotated normal vector fields in Bézier form)

By Theorem 1, given two polynomials $u(t)$ and $v(t)$ of degree μ, we can express the rotated normal vector field of a PH curve in Bernstein Bézier form as: $V(t) = \sum_{k=0}^{2\mu} B_k^{2\mu}(t) q_k$ where the control points q_k are $q_k = [-J_k, I_k, H_k]$, for $k = 0 \ldots 2\mu$, and the I_k and J_k are as in the Theorem 10 in the appendix. The $B(t)$ and $V(t)$ of Definition 4 define the ruled surface $S(s,t) = B(t) + sV(t)$ which has inclination angle of 45 degrees against the *xy*-plane. The ruled surface is the γ-map $\gamma\big(B(t)\big) = B(t) + sV(t) = [x(t) - sy'(t), y(t) + sx'(t), s\sigma(t)]$.

For a planar PH curve $B(t)$ on the z=0 plane, every point (x, y, z) of the rotated normal vector field of $B(t)$ is associated with a cycle tangent to $B(t)$ on the $z = 0$ plane that is oriented compatible with the curve. The cyclographic map for a set of 2D boundaries is a set of ruled surfaces. We can find the Medial Axis Transform (MAT) of those boundaries by restricting the ruled γ-map surfaces to the segments of the generators between the $z = 0$ plane and the first singularity of the surface above the $z = 0$ plane. Then the end points of the segments are points on the MAT [2]. The MAT for a PH curve $B(t)$ with a ray can be found in general by intersecting a ruled surface and a plane; the MAT for a PH curve $B(t)$ with a cycle similarly can be transformed to the problem of intersecting a ruled surface and a cone.

2.3 The LP and CP Problem

Theorem 4: (Ruled surface and plane intersection) [22]

Assume that the ruled surface $S(s,t) = B(t) + sV(t)$ has the base curve $B(t) = \sum_{i=0}^{n} B_i^n(t) p_i$ and ruling vectors $V(t) = \sum_{i=0}^{m} B_i^m(t) q_i$ that are Bézier curves of degree n and m and which have the control points p_i, $i = 0 \ldots n$, and q_i, $i = 0 \ldots m$. The intersection curve of the ruled surface and the plane $Ax + By + Cz + D = 0$ is a rational Bézier curve $C(t) = (\sum_{i=0}^{m+n} w_k B_k^{m+n}(t) r_k)/(\sum_{i=0}^{m+n} w_k B_k^{m+n}(t))$ of degree $m + n$.

Theorem 5: (Intersection of an implicit cone $x^2 + y^2 - (z + r)^2 = 0$ and the γ-map of a PH curve) [5],[6],[21]

Consider a ruled surface $S(s,t) = B(t) + sV(t)$ whose base curve $B(t) = \sum_{i=0}^{n} B_i^n(t) p_i$ is a planar PH curve in the *xy*-plane, in Bézier form of degree n, with control points p_i, $i = 0, \ldots, n$. Assume that the ruled vector field $V(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t) q_i$ is the rotated normal vector field of $B(t)$. Let the control points for $V(t)$ be q_i, $i = 0, \ldots, n-1$. By theorems 1 to 3, q_i can be derived from $B(t)$. Then the *xyz* components of the ruled vector field form the Pythagorean triple $\big(V^x(t)\big)^2 + \big(V^y(t)\big)^2 = \big(V^z(t)\big)^2$ where the superscript designates the coordinates. Moreover, the ruled surface intersects the implicit cone $x^2 + y^2 - (z - r)^2 = 0$ in the rational Bézier curve $C(t) = (\sum_{k=0}^{3n-1} w_k B_k^{3n-1}(t) r_k)/(\sum_{k=0}^{3n-1} w_k B_k^{3n-1}(t))$ of degree $3n-1$, with the weights w_k and the control points r_k are described, for $k = 0, \ldots, 3n-1$, by:

$$
\begin{gathered}
w_k = b_k^{(n)} \\
r_k = \Big(\sum_{i=0}^{k} \big(\varphi_{i,k-l}^{2n-1,n} b_i p_{k-i} + \varphi_{i,k-i}^{2n,n-1} c_i q_{k-i}\big)\Big)/b_k^{(n)} \\
\mathrm{b_k} = \sum_{\mathrm{i=0}}^{\mathrm{k}} \varphi_{\mathrm{i,k-i}}^{\mathrm{n,n-1}}(-2) \begin{bmatrix} \mathrm{p_i^x} & \mathrm{p_i^y} & \mathrm{r} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{q_{k-i}^x} & \mathrm{q_{k-i}^y} & \mathrm{q_{k-i}^z} \end{bmatrix}, \mathrm{k} = 0, \ldots, 2\mathrm{n}-1, \\
\mathrm{c_k} = \sum_{\mathrm{i=0}}^{\mathrm{k}} \varphi_{\mathrm{i,k-i}}^{\mathrm{n,n}} \begin{bmatrix} \mathrm{p_i^x} & \mathrm{p_i^y} & \mathrm{r} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{q_{k-i}^x} & \mathrm{q_{k-i}^y} & -\mathrm{r} \end{bmatrix}, \mathrm{k} = 0, \ldots, 2\mathrm{n}, \\
\mathrm{b_k^{(n)}} = \sum_{\mathrm{j=0}}^{\mathrm{2n-1}} \sum_{\mathrm{i=0}}^{\mathrm{j}} \varphi_{\mathrm{j,k-j}}^{\mathrm{2n-1,n}} \varphi_{\mathrm{i,j-i}}^{\mathrm{n,n-1}} (-2) \begin{bmatrix} \mathrm{p_i^x} & \mathrm{p_i^y} & \mathrm{r} \end{bmatrix} \cdot \begin{bmatrix} \mathrm{q_{j-i}^x} & \mathrm{q_{j-i}^y} & \mathrm{q_{j-i}^z} \end{bmatrix}
\end{gathered}
\tag{3}
$$

Example 1: (The CP problem)

Let C be the cycle $x^2 + y^2 - (-10)^2 = 0$ in the $z = 0$ plane. The oriented radius is -10, $\gamma(C)$ is the cone $x^2 + y^2 - \big(z - (-10)\big)^2 = 0$. For the planar PH curve $B(t)$, whose initial values are $(u_0, u_1) = (10,3)$, $(v_0, v_1) = (2,-1)$, $p_0 = (-20,-30,0)$, we obtain $p_1 = \left(12, -\frac{50}{3,0}\right)$, $p_2 = \left(\frac{68}{3}, -18,0\right)$, $p_3 = \left(\frac{76}{3}, -20,0\right)$, and

$q_0 = (-40,96,104),\ q_1 = (4,32,28),\ q_2 = (6,8,10)$. The MAT for $B(t)$ and C is the intersection curve of $\gamma(B(t))$ and $\gamma(C)$. By Theorem 5, the MAT is a rational Bézier curve of degree 8, and its control points r_k and the associated weights w_k are listed in Table 1. Fig. 1 shows the circle and PH curve in 2D on the left with the associated bisector. To the right we see the cyclographic maps of the circle and the PH curve. Their intersection projects to the bisector curve.

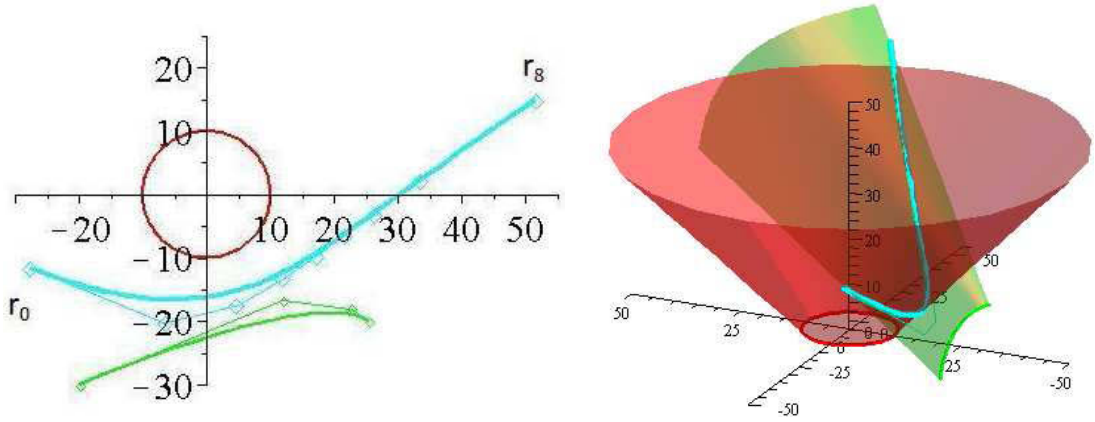

Fig. 1: The MAT of PH curve $B(t)$ and circle C.

control points	weights
$r_0 = [-360/13, -150/13, 20]$	$w_0 = 6240$
$r_1 = [-592/89, -1798/89, 348/89]$	$w_1 = 5340$
$r_2 = [68440/15359, -270018/15359, 44828/15359]$	$w_2 = 30718/7$
$r_3 = [426480/36149, -481710/36149, 196780/36149]$	$w_3 = 72298/21$
$r_4 = [1146612/67003, -647422/67003, 573392/67003]$	$w_4 = 268012/105$
$r_5 = [395947/18456, -40207/6152, 217081/18456]$	$w_5 = 12304/7$
$r_6 = [893894/34329, -35600/11443, 536494/34329]$	$w_6 = 22886/21$
$r_7 = [85291/2571, 5737/2571, 57821/2571]$	$w_7 = 1714/3$
$r_8 = [4171/81, 3616/243, 10595/243]$	$w_8 = 216$

Tab. 1: The control points and weights for Example 1.

3 THE ALGEBRA OF PPP, PPC AND PPL

3.1 The Algebra of P

Let $[X, Y, Z]$ be the points on the cyclographic map of a PH curve with the base curve $[x(t), y(t), 0]$ and vector field $[-y'(t), x'(t), \sigma(t)]$. To simplify the notation, we just write $[x, y, 0]$ and $[-y', x', \sigma]$ for the base curve and the vector field. By eliminating the s variable in equation $[X, Y, Z] = [x, y, 0] + s[-y', x', \sigma]$, we obtain $[X, Y] = [x, y] + Z \cdot [-y'/\sigma, x'/\sigma]$. Note that if Z is fixed, this equation is the offset of the base curve, and if t is fixed, then the equation generates the normal line of the base curve at $[X, Y]$.

3.2 The Implicit Form of the Cyclographic Map of P

Theorem 6 [22]**:** Assume the ruled surface $S(s, t_1) = (1 - s)B_1(t_1) + sB_2(t_1)$ is generated by two space curves $B_1(t_1)$ and $B_2(t_1)$ of degree m and n, respectively. Then the implicit form of S has degree $m + n$.

The curves $B_1(t_1)$ and $B_2(t_1)$ are called the *directrices*. Consider the cyclographic map for the polynomial PH Bézier curve $B(t)$. Since the curve is PH, the rotated vector field $V(t)$ is also polynomial. We use $B_1(t_1) = B(t_1)$ and $B_2(t_1) = B(t_1) + V(t_1)$, as the two directrices of the cyclographic map $B(t) + sV(t)$. Both curves have degree n, and there are no base points because the homogeneous variable is always 1. So, from Sederberg's paper, the implicit form has degree at most $2n$. Can the implicit form have a lower degree? We prove the following theorem:

Theorem 7: The implicit form of the cyclographic map of a degree n PH curve has degree $2n-1$.

Proof: In [22], Sederberg generates a family of planes $L(t)$, each plane containing the two directrix points $B_1(t) = B(t)$ and $B_2(t) = B(t) + sV(t)$. $L(t)$ can thus be said to follow the generators of $S(s,t)$. By finding two independent planes of the family $L(t)$, he generates a system of equations via $B_1(t) \cdot L(t) = 0$ and $B_2(t) \cdot L(t) = 0$, and derives a matrix with $n_0 + n_1 + 2m + 2$ rows and $4m + 4$ columns, where m is the degree of $L(t)$, n_i is the degree of $B_i(t)$, $i = 1,2$. After this process, he finds the implicit form with degree $n_0 + n_1$. For the implicitization of the cyclographic map of PH curves, we can generate the system of equations via $B(t) \cdot L(t) = 0$ and $[B(t) + V(t)] \cdot L(t) = 0$, which is equivalent to generating the system of equations for $B(t) \cdot L(t) = 0$ and $V(t) \cdot L(t) = 0$. By Sederberg's work, the degree of the implicit form is the sum of the degrees of $B(t)$ and of $V(t)$, which is $2n-1$.

♠

Note that the plane $L(t)$ passing through $B(t)$ has a normal perpendicular to $V(t)$, which is also perpendicular to the generators of $S(s,t)$.

To determine the degree of the cyclographic map, we can compute the implicit form. We find the Gröbner Basis for the polynomial set $F: [X - B_{x(t)} - sV_x(t), Y - B_y(t) - sV_y(t), Z - sV_z(t)]$ w.r.t. the lexicographic order with $s > t > X > Y > Z$. The Gröbner Basis of F will then contain the implicit form of the cyclographic map of the PH curves.

Example 2: Consider the PH curve whose initial value are $(u_0, u_1) = (1,5)$, $(v_0, v_1) = (5,1)$, $p_0 = (18, -52/3)$. Then the implicit form of its cyclographic map is of degree 5:

$$
\begin{aligned}
F(X,Y,Z,W) = {} & 144(X^5-3X^4Z+X^3Y^2+2X^3Z^2-3X^2Y^2Z+2X^2Z^3+3XY^2Z^2-3XZ^4-Y^2Z^3+Z^5) \\
& +(-21600X^4+2784X^3Y+36288X^3Z-34344X^2Y^2-8352X^2YZ+22032XY^2Z+8352XYZ^2 \\
& \quad -22464XZ^3-17496Y^4+12312Y^2Z^2-2784YZ^3+7776Z^4) \\
& +(1186984X^3-663984X^2Y-851016X^2Z+1155708XY^2+425952XYZ-241512XZ^2 \\
& \quad -676512Y^3+34020Y^2Z+238032YZ^2+278792Z^3) \\
& +(-31644144X^2+22343688XY+6934320XZ-19335510Y^2+657720YZ+4507776Z^2) \\
& +(422619957X-247387284Y+12440925Z-2339604810)
\end{aligned}
$$

3.3 The Algebra of PP

To simplify the description, we denote all of functions of the base curve and vector field without their parameter. That is, x_i means $x_i(t_i)$, and the notation for, $B_i, V_i, y_i, \sigma_i, x'_i, y'_i$ is analogous.

Theorem 8 [12]: Let $B_i = (x_i, y_i, 0), i = 1,2$, be two base curves and let $V_i = (-y'_i, x'_i, \sigma_i)$ be the vector fields of the cyclographic maps of two PH curves. That is, $\gamma(B_i) = B_i + s_{iV_i}$, $i = 1,2$. Then the intersection of these two surfaces satisfies:

$$f(t_1,t_2) = \begin{vmatrix} B_2 - B_1 \\ V_1 \\ V_2 \end{vmatrix} = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 & 0 \\ -y_1' & x_1' & \sigma_1 \\ -y_2' & x_2' & \sigma_2 \end{vmatrix} = 0 \tag{4}$$

Furthermore, the parametric form of the intersection points $[X, Y, Z]$ under the above assumptions is:

$$[X(t_1,t_2), Y(t_1,t_2), Z(t_1,t_2)] = [x_1, y_1, 0] + \frac{\sigma_2(y_2 - y_1)}{\begin{vmatrix} x_1' & x_2' \\ \sigma_1 & \sigma_2 \end{vmatrix}} [-y_1', x_1', 1] \tag{5}$$

Proof: The theorem can be proved easily by the algebraic property of PH curves, namely $[X,Y] = [x_1, y_1] + Z\left[-y_1'/\sigma_1, x_1'/\sigma_1\right] = [x_2, y_2] + Z\left[-y_2'/\sigma_2, x_2'/\sigma_2\right]$. The Z value can be found by eliminating X and Y, and the intersection of these two surfaces satisfies $f(t_1, t_2) = 0$, where $f(t_1, t_2) = 0$ can be found by eliminating Z. After finding Z, X and Y can be derived by $[X,Y] = [x_1, y_1] + Z[-y_1'/\sigma_1, x_1'/\sigma_1]$.

♠

Note that the vectors, $B_2(t_2) - B_1(t_1)$, $V_1(t_1)$, $V_2(t_2)$ are linearly dependent, thus lie in the same plane.

3.4 The Algebra of LPP and CPP

We know the implicit and parametric forms of $\gamma(L), \gamma(C)$ and $\gamma(P)$. We know the parametric form of $\gamma(L) \cap \gamma(P)$ (Theorem **4**), and of $\gamma(C) \cap \gamma(P)$ (Theorem 5). We also know the parameter relationship of $\gamma(P_1) \cap \gamma(P_2)$ (Theorem 8). Now we would like to find $\gamma(L) \cap \gamma(P_1) \cap \gamma(P_2)$ and $\gamma(C) \cap \gamma(P_1) \cap \gamma(P_2)$, so that we can determine all variable-radius circles tangent to a line or circle and two PH curves.

Consider the LPP problem. Denote the three entities with L, P_1 and P_2, where we may assume that the degree n_1 of P_1 is not greater than the degree n_2 of P_2. Assume that t_i is the parameter of the base curve of P_i. We can find $\gamma(L) \cap \gamma(P_1) \cap \gamma(P_2)$ as follows:

1. Generate the intersection curve $\zeta = \gamma(L) \cap \gamma(P_1)$.
2. Substitute the parametric form ζ into the implicit form of $\gamma(P_2)$.

The degree of parametric form of ζ is $2n_1 - 1$, and the implicit form of $\gamma(P_2)$ is degree $2n_2 - 1$, so we generated a degree $(2n_1 - 1)(2n_2 - 1)$ equation with t_1 as its variable.

Assume the degree for the PH curve is n in the LLP and LCP case, and the degrees are $n_1, n_2 (n_1 \leqq n_2)$ for the P_1, P_2 respectively in the CPP case. Proceeding similarly, we can find one equation with one variable for LLP, LCP, and CPP, with degree $2n - 1$, $3n - 1$ and $(3n_1 - 1)(2n_2 - 1)$, respectively. Assume all of the PH curves mentioned above have degree 3; then we need to solve univariate equations of degree 5, 8, 25, and 40 for the LLP, LCP, LPP, CPP cases, respectively. Especially the LPP and CPP cases generate polynomials that would be difficult to solve with reasonable accuracy.

	L_1L_2P	$LCP(\&CCP)$	CP_1P_2 $(n_1 \leqq n_2)$
Parametric Curve ζ Degree of ζ	$\zeta = \gamma(L_1) \cap \gamma(P)$ $2n - 1$	$\zeta = \gamma(C) \cap \gamma(P)$ $3n - 1$	$\zeta = \gamma(C) \cap \gamma(P_1)$ $3n_1 - 1$
Substitute into // degree	$\gamma(L_2)$ // 1	$\gamma(L)$ // 1	$\gamma(P_1)$ // $2n_2 - 1$
Total Degree	$2n - 1$	$3n - 1$	$(3n_1 - 1)(2n_2 - 1)$
Equation variable of:	P	P	P_1

Tab. 2: The polynomial degrees for LLP, LCP, CCP and CPP.

There are other ways to solve the problem reaching the same degrees. For example, for the LLP case, we can generate the parametric form for the line generate by $\gamma(L_1) \cap \gamma(L_2)$ and substitute it into the implicit form of $\gamma(P)$. Then, the degree for the univariate equation is $2n - 1$.

Example 3 [CPP problem]: Consider the circle C and one PH curve P_1, the entities of Example 1, and the PH curve P_2 defined in Example 2. We can find the circle tangent to these three geometric entities by intersecting their cyclographic maps. We plug the intersection curve found in Example 1 into the implicit form found in Example 2, and obtain a degree 36 equation with one variable. After solving this equation, it takes less one second in Maple to find 8 real solutions, -1.489, 0.386, 1.089, 1.090, 1.503, 1.547, 2.380, 4.139. But there is only one solution in the range from 0 to 1, so we obtain just one circle with radius 6.018, centered at (6.906,-14.453), as shown in Fig. 2. Again, the spatial image shows the intersecting cyclographic maps.

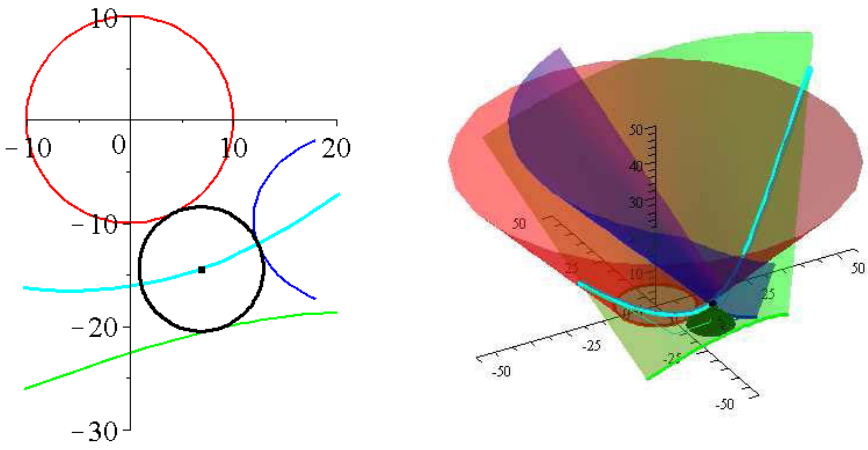

Fig. 2: The CPP problem of Example 3. The Apollonius circle is shown in black.

Note that if the circle is not centered at the origin, we can always translate all three geometric entities, so that the center of the circle is at the origin. After we find the solutions, the inverse translation recovers the true solution. The translation of the p_0 in the PH curve translates the whole PH curve since the other control points are derived from p_0.

3.5 The Algebra of PPP

Theorem 9: Let $B_i = (x_i, y_i, 0), i = 1,2,3$, be two base curves and let $V_i = (-y_i', x_i', \sigma_i)$ be the vector fields of the cyclographic maps of two PH curves. The intersection of these two surfaces satisfies:

$$\begin{aligned} f_1(t_1,t_2) &= \begin{vmatrix} x_2-x_1 & y_2-y_1 & 0 \\ -y_1' & x_1' & \sigma_1 \\ -y_2' & x_2' & \sigma_2 \end{vmatrix} = 0 \\ f_2(t_1,t_3) &= \begin{vmatrix} x_3-x_1 & y_3-y_1 & 0 \\ -y_1' & x_1' & \sigma_1 \\ -y_3' & x_3' & \sigma_3 \end{vmatrix} = 0 \\ f_3(t_1,t_2,t_3) &= \begin{vmatrix} 0 & x_1' & \sigma_1 \\ (y_1-y_2)\sigma_2 & x_2' & \sigma_2 \\ (y_1-y_3)\sigma_3 & x_3' & \sigma_3 \end{vmatrix} = 0 \end{aligned} \tag{6}$$

Furthermore, the parametric form of the intersection points $[X, Y, Z]$ under the above restriction is:

$$[X(t_1,t_2), Y(t_1,t_2), Z(t_1,t_2)] = [x_1, y_1, 0] + \frac{\sigma_2(y_2-y_1)}{\begin{vmatrix} x_1' & x_2' \\ \sigma_1 & \sigma_2 \end{vmatrix}} [-y_1', x_1', 1] \tag{7}$$

Proof: The theorem can be proved by using the equations $[X, Y] = [x_1, y_1] + Z[-y_1'/\sigma_1, x_1'/\sigma_1] = [x_2, y_2] + Z[-y_2'/\sigma_2, x_2'/\sigma_2] = [x_3, y_3] + Z[-y_3'/\sigma_3, x_3'/\sigma_3]$. The equations $f_1(t_1, t_2)$ and $f_2(t_1, t_3)$ can be found by eliminating X, Y, Z from the equations

$[x_1, y_1] + Z[-y_1'/\sigma_1, x_1'/\sigma_1] = [x_2, y_2] + Z[-y_2'/\sigma_2, x_2'/\sigma_2]$
$[x_1, y_1] + Z[-y_1'/\sigma_1, x_1'/\sigma_1] = [x_3, y_3] + Z[-y_3'/\sigma_3, x_3'/\sigma_3]$

respectively. Here $f_3(t_1, t_2, t_3)$ expresses that the two Z values in the above two equations should be equal.

♠

Assume the degrees of P_1, P_2 and P_3 are n_1, n_2 and n_3 respectively, where $n_1 \le n_2 \le n_3$. We can find the intersection $\gamma(P_1) \cap \gamma(P_2) \cap \gamma(P_3)$ by:

1. Substituting the parametric form $\gamma(P_3)$, into implicit form of $\gamma(P_1)$,
2. Substituting the parametric form $\gamma(P_2)$, into implicit form of $\gamma(P_1)$,
3. Solving the two equations with two variables so obtained, restricting to solutions that lie in the domain $0 \leqq s_3, t_3 \leqq 1$.

We generate two bivariate equations in (s_3, t_3) of degree $2n_1n_2$ and $2n_1n_3$ respectively in the first and second step. As example, assume that the degree of all PH curves is 3. So we need to solve two degree 15 equations with 2 variables in this procedure. There are at most 225 solutions, but many of them will lie outside the domain of interest.

If we intersect the implicit form of these 3 surfaces, however, then there are at most 125 solutions. In this case, we need to solve 3 equations with 3 variables. Both ways of finding solutions for the PPP problem are not attractive because the systems of equations have high algebraic degree and so are not easy to solve. So we explore alternative solutions in the next section.

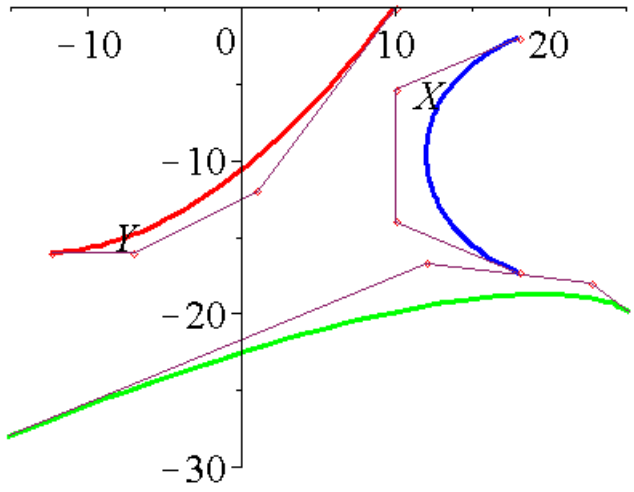

Fig. 3: The PPP case. The control points are given in the table.

	(u_0, u_1)	(v_0, v_1)	p_0	p_1	p_2	p_3	In Example
P_1	$(-3,0)$	$(6,4)$	$[10,0]$	$[1,-12]$	$[-7,-16]$	$[-37/3,-16]$	
P_2	$(10,3)$	$(2,-1)$	$[-20,-30]$	$[12,-50/3]$	$[68/3,-18]$	$[76/3,-20]$	Example 1
P_3	$(1,5)$	$(5,1)$	$[18,-52/3]$	$[10,-14]$	$[10,-16/3]$	$[18,-2]$	Example 2

Tab. 3: Control points of the curves in Fig. 3.

4 HARDWARE-ASSISTED SOLUTIONS

4.1 Approach

As we have seen, the equations obtained by the algebraic derivation have high degrees. This means that solving them is both computationally demanding and numerically delicate. Moreover, if we are to restrict to solutions in the parametric range [0,1], or any other domain, many of the equation roots obtained algebraically will not qualify, implying wasted computations. For example, we expect only one common circle tangent to the three curves in Fig. 3, yet the algebraic degrees of the two equations in (s_3, t_3) are 15 each, so that in the worst case up to 225 roots have to be determined and examined to find the one root that has geometric significance. Because of such considerations, we investigate a solution method that is both quick and does not have to find solutions that lie outside the region of interest. We describe such a method next that exploits the highly parallel nature of sampling and the capability of graphics hardware to evaluate such samplings rapidly. Briefly, we will render the

tessellated cyclographic maps by hardware and use the *z*-buffer to extract intersections of the various surfaces. This is not a new idea, and an example of using this approach to compute Voronoi diagrams of points, line segments and polygons goes back to [16]. Since, an extensive literature has sprung up to accelerate Voronoi computations using the GPU; see, e.g., [20], [23].

Our approach is to approximate the cyclographic map by a tessellation of the appropriate accuracy, render the approximations in a raster of, say, 1000x1000 pixels, and then extract the intersection points from the graphics hardware. Those points can then be used directly, or else we can use them as starting points for further refinement such as Newton iteration or other suitable numerical procedures. Because of the geometry of the cyclographic map, tessellation is quite straightforward and error bounds are not difficult to obtain for its accuracy.

We recall that the cyclographic maps of points, oriented circles and oriented lines are right-circular cones and planes. They are easy to approximate with triangular and quadrilateral facets. Following [16], we can tessellate a right-circular cone with approximately 100 triangles assuming a raster of 1000x1000 and achieve sub-pixel accuracy. Tessellating PH curves can be done exactly as described in Hoff [16].

Briefly, the discretization of Hoff is to treat consecutive segments as line segments, each giving rise to a quadrilateral facet that is delimited by the normals of the line segments at their ends. The triangular gaps that arise between consecutive segments on one side are closed by pieces of cones with apex at the joining vertex. The situation is illustrated in Fig. 4 (left). Our discretization is as follows. After an initial segmentation of the curve, we evaluate the curve normals at the segment end points. If two consecutive normals diverge by more than 3.6 degrees (adopting the same discretization accuracy as for the circular cone), the segment is subdivided by a curve point in the middle, into two segments. The resulting quadrilaterals may be twisted, and we divide them into two triangles (Fig. 4, right). Subdivision is repeated until each consecutive normal turns no more than 3.6 degrees. Our method generates fewer triangles, but we have not studied whether this confers a performance improvement.

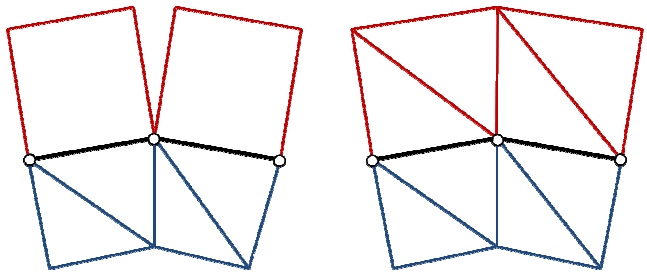

Fig. 4: Base curve in black, two segments shown. *Left:* Hoff et al tessellation of the cyclographic map; main quadrilaterals, in red, delimited by perpendiculars to the segment, and gaps on the convex side are filled with cone approximations. *Right:* Our tessellation uses rotated curve normals to delimit skew quadrilaterals which are then triangulated as shown.

Figure 5 shows the distance function rendered by the graphics hardware for the configurations of Fig. 2, the CPP case, and for Fig. 3, the PPP case. The time to determine each is approximately 0.025sec using an NVIDIA Quadro FX 1700 graphics card. Control points can be picked and moved at 41 frames per second. On a GeForce GTX 280 we achieved over 500 frames/second, i.e., determined the solution in less than 0.002sec. We use the two-sided distance function in the figure.

4.2 Basic Algorithm and Results

Given the three planar shape elements for which we wish to solve the Apollonius problem, we discretize them into connected segments. The facets of the cyclographic map are generated as explained. All facets belonging to one of the shape elements are colored with a specific color, in our examples red, green and blue. They are then rendered using flat shading. A shader code fragment is applied next that notes all pixels adjacent to a differently colored pixel and colors it white, so

accentuating the medial axis of two of the three shapes. Moreover, pixels adjacent to two or more other colors are the sought solution points and are likewise noted. The coordinates of the intersections are printed to the console, but could be used instead for a subsequent refinement computation. Assuming a raster of 1000x1000 pixels and a normal separation of no more than 3.6 degrees, we expect to obtain the points with a relative accuracy of the results of 10^{-3}, a good initial guess for a possible subsequent Newton iteration.

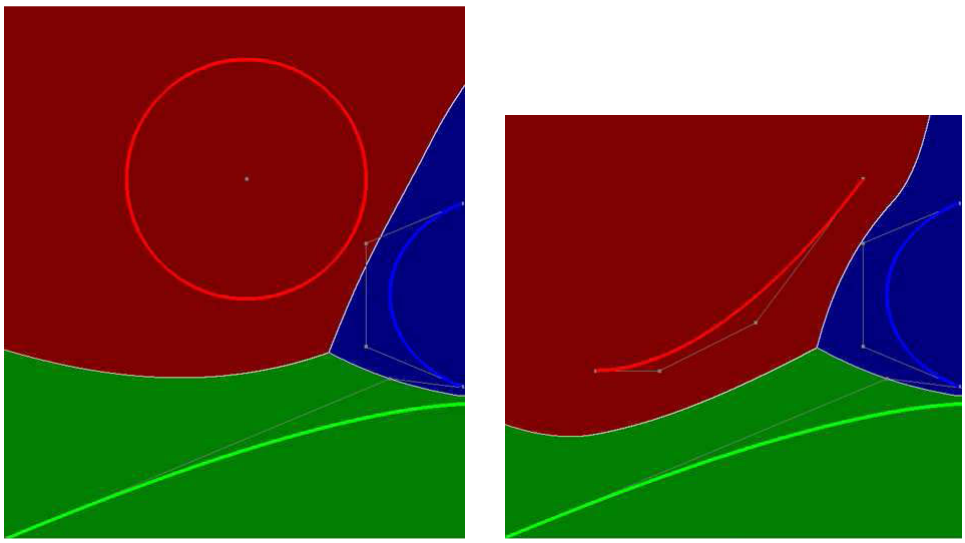

Fig. 5: CPP and PPP configurations from Figures 2 and 3. The control poly-arcs of the PH curves are drawn in light gray.

In the examples we show, the two-sided cyclographic map is used. That is, a shape element is considered in both direction and only the positive part of the cyclographic map is used. End points are capped with half cones. When using the one-sided distance function only, there are undefined distance regions outside the parametric domain, past the end of a PH curve, and no end caps. Likewise, the interior of circles has no distance information, or the exterior does not, depending on the circle orientation. Since the signed, negative distance function must be clipped, those regions reveal the interaction of the positive part of the cyclographic maps of other shape elements beyond the distance function. An example is shown in Fig. 6 on the left.

In the algebraic derivation of Section 3 we work with the underlying algebraic curve when deriving the implicit representation of the cyclographic map of a PH curve. This is one of the reasons why the algebraic approach delivers more solutions than are geometrically meaningful. An example of the extended cyclographic map is shown in Fig. 6 on the right. Note the extraneous solution marked by the yellow circle. Here, an Apollonius circle will touch the green curve outside the [0,1] domain.

The hardware assisted approach is very general owing to the fact that once a segmentation of the geometric entity is obtained, the approximate cyclographic map is readily constructed and evaluated by the hardware. Thus, there is no need to restrict to PH curves. Furthermore, for most plane curves and splines there are good algorithms for piecewise-linear approximations and normal evaluation; e.g., [8]. Therefore, all those curves can be used as entities for Apollonius constructions. The details are routine. Likewise, there is no reason to restrict the map construction to three shape elements at a time. We can easily create the distance maps for multiple objects and, at point-intersections, we can construct circles touching a triple of those elements. Here we assume that there are no degenerate intersections of multiple bisectors. An example of multiple objects is shown in Fig. 7.

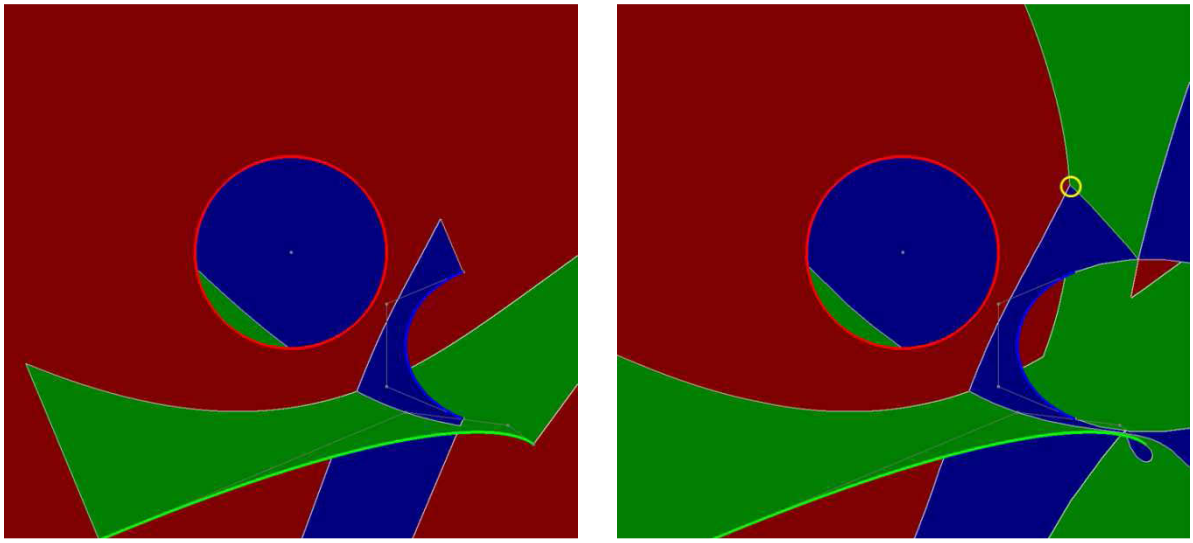

Fig. 6: One-sided distance function. Objects are from Fig. 3. *Left*: cyclographic map restricted to PH curve domain [0,1]. *Right*: map for extended parameter domain [−10,4].

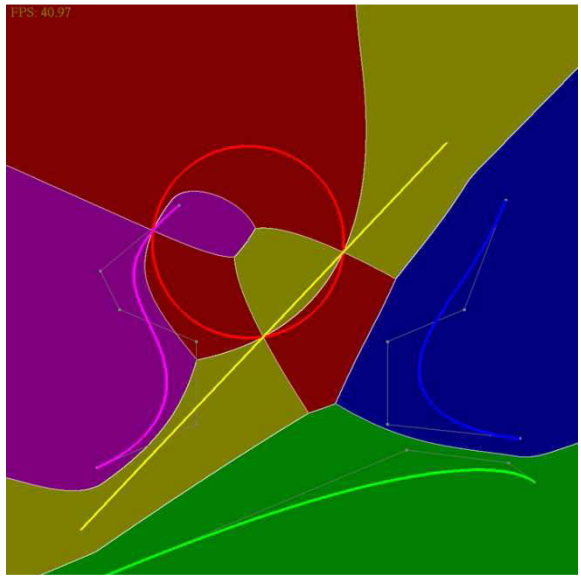

Fig. 7: Distance maps of multiple objects including non-PH Bézier curves.

The accuracy obtained by this approach to determining circles tangent to given entities depends not only on the accuracy of the tessellation but also on the condition number of intersecting facets. Briefly, when the facets of two different shape elements are nearly parallel, then the accuracy of determining their intersection declines. This is primarily a characteristic of the problem, not of the method. However, as noted in [16], the graphics hardware determines the intersection by a depth sampling using single-precision floating-point numbers. Inaccuracies are visible in a large raster but are usually subtle in appearance. Fig. 8 shows an example: The control points are symmetric about the $y = -1$ axis and the two curves meet with tangent continuity at the point $(-1,-1)$. Thus, the

cyclographic maps should intersect in the line of symmetry. The actual intersection, however, shows a slight downward slope.

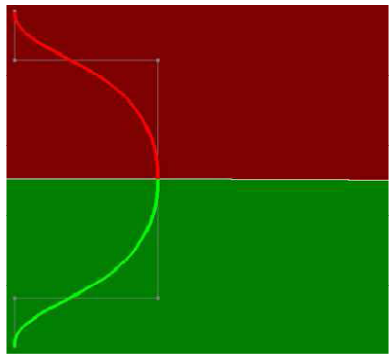

Fig. 8: Ill-conditioned intersection of the cyclographic maps. The red and green curves are symmetric, yet the intersection of the cyclographic maps is not quite.

4.3 Medial Axis Transform

The *medial axis transform* (MAT), e.g., [2], is a restriction of the cyclographic map and can be evaluated using the same approach. However, so far we have exploited the fact that the intersections are between the cyclographic maps of different geometric elements and thus can be recognized from color discontinuities of adjacent pixels. When the MAT of a single entity is to be determined, however, the self-intersections must be found. Thus, there is no color discontinuity. Moreover, near centers of curvature the intersections become ill-conditioned. So, we proceed as follows.

We render the tessellated cyclographic map to the *depth buffer* of the graphics card. Then, we identify the gradient of the depth at each pixel. The MAT points are identified by pixels with a large variance in their gradient and the gradient of their neighbors. Fig. 9 shows the gradient image. The shader code fragment calculates the pixel gradients in a first pass, and then compares gradient values in a second pass. Fig. 10 shows an example of the MAT.

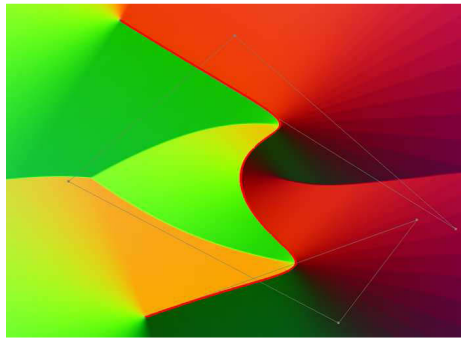

Fig. 9: MAT as gradient discontinuities.

4.4 Multiple Sampling and Surface Decomposition

For Voronoi diagrams, medial axis and general distance maps, a single rendered image suffices, a fact that is often assumed in the literature. Constraint solving, however, has additional requirements if all geometrically meaningful solutions should be considered. Those additional requirements can be discharged by collecting results from multiple raster renderings, each accounting for a family of solutions, and decomposition of the cyclographic map in some of these cases. We illustrate the situation with the classical Apollonius problem and sketch our approach how to deal with it.

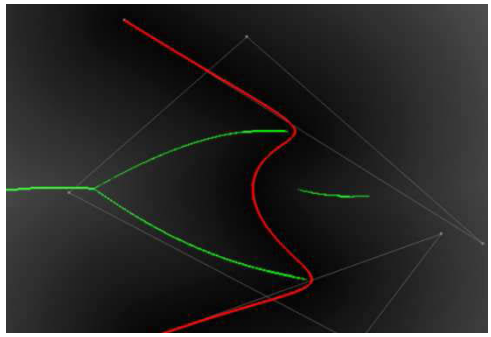

Fig. 10: Extracted MAT of a single shape.

It is well-known that the classical Apollonius problem can be factored into 4 systems of equations, each requiring solving only a univariate quadratic equation; e.g., [18]. This can be understood by considering two cycles and asking for the locus of all cycles tangent to the two given ones. This locus is a conic section, and the different systems are based on different orientations of the three cycles which affects the cyclographic maps that must be intersected.

We observe that we need to account for different cases arising from orientation. However, this is not enough: If we are to find all solutions to the Apollonius problem using hardware assistance, as described, then we need to be able to generate the conics of two cycles in their entirety. The problem that arises is that some parts of the conic do not correspond to the global bisector, as illustrated in Fig. 11. Here we are looking for cycles tangent to a positive cycle (green) and a negative one (red). The blue cycle is an example of such a cycle, but the part of the hyperbolic arc on which its center is found is obscured by the lower part of the green cone. Thus, we cannot find all solutions of the Apollonius problem without separate rasters rendering the sub-cases. See also Fig. 12 (left).

We can approach this situation by decomposing the cyclographic map of (positive) cycles into two parts: the lower cone, representing the interior distance function, extending from the perimeter to the vertex above the center, and the upper cone, beginning at the vertex above the center at height equal to the radius and extending upward to infinity. Since the rendering process is so fast, multiple renderings are an attractive solution. An alternative to decomposition, based on the geometry of cyclographic maps, is *depth buffer reversal*. Briefly, in ordinary rendering the geometric shape closest to the eye is considered visible and rendered. This is done by comparing the depth of competing shape elements and selecting the nearest one. In depth buffer reversal the farthest shape element is rendered instead. In Fig. 12, on the right, the effect of depth buffer reversal is shown for the same problem instance. Thus an alternative, for the classical Apollonius problem, would be to render the raster twice, with and without depth buffer reversal. Note that this problem may also arise for circular arcs and Bézier curves at areas of high curvature.

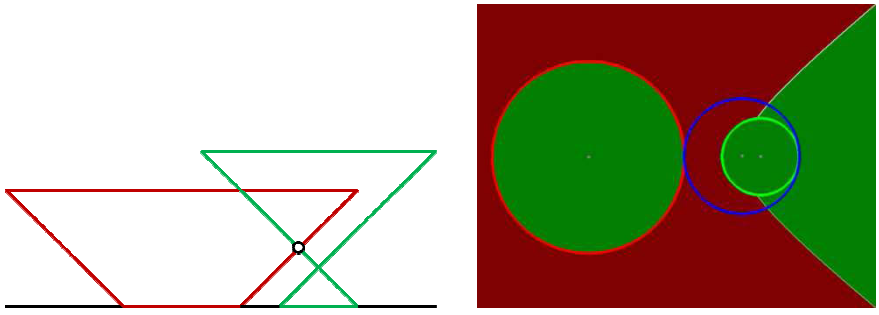

Fig. 11: Hyperbolic locus of the centers of all cycles tangent to negative red and positive green cycles. Note the missing arc on which the center of the blue example solution lies.

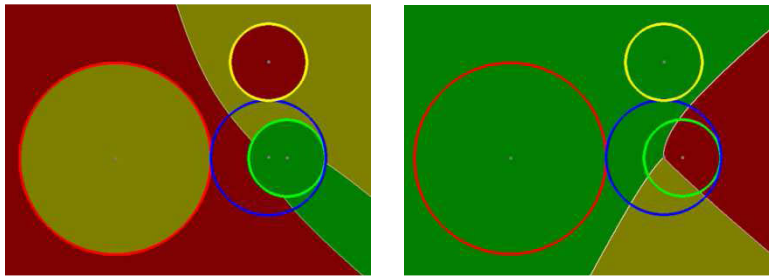

Fig. 12: *Left*: Instance of Apollonius problem requiring decomposition of cyclographic map. *Right*: Depth buffer reversal as an alternative to decomposition for circles.

5 SUMMARY AND DISCUSSION

We have reviewed algebraic solutions to the generalized Apollonius problem involving lines, circles and PH curves. The simpler cases were discussed in [5],[6], the more complex cases here. The use of PH curves drives up the algebraic degree of the equation systems that arise. Therefore, solvers constructed based on the algebra have to cope with an increasing computational and algorithmic burden as the number, and degrees, of the PH curves increases.

For high-degree systems, the number of solutions of the equation system becomes large and results in much unproductive computation. For this reason, we have explored an alternative in which the geometric objects are each represented by a poly-line, so that hardware acceleration can be used, utilizing commodity graphics hardware.

All experiments have been done on a PC with an Intel quad-core Xeon CPU of 3.16GHz, 4GB main memory. With an NVIDIA Quadro FX 1700 graphics card, all graphics-accelerated computations achieve rates of approximately 41 frames per second, and with an NVIDIA GeForce GTX 280 a speedup of better than an order of magnitude is obtained, with frame rates around 500 fps, or 2 msec. At these speeds a purely algebraic solution becomes unattractive. This finding is not unlike the results in the literature on GPU algorithms. Here, we refer the reader to the survey [20] and for (approximate) distance field computations in 3D, a close relative to the medial axis, to [23].

A key advantage of the hardware assisted approach is that the repertoire of geometric objects can be greatly increased at little or no extra cost. Thus, there is no need to restrict to PH-curves, since the ability to construct easily implicit and parametric forms for their cyclographic maps is not important to their tessellation. However, the limitations of the GPU approach concern the ability to resolve at sub-pixel scales. With the chosen raster size of 1000 by 1000, we should not expect more than three digits accuracy. Such accuracy is reasonable for quick, interactive exploration, but not for accurate results that may be needed by applications such as CAD. So, for better accuracy a stable numerical method could be used to complement the GPU computation. That is, run the GPU computation to get approximate solutions that are then refined by subsequent numerical computation.

An alternative to the mixed GPU/iteration would be a multi-resolution approach: Having found an approximate solution, one can restrict the region of sampling and resample the smaller region at high resolution, so refining the solution estimates. This approach has many attractive sides, among them that such a “zoom” can be executed without necessarily reading back the graphics buffers into main memory, a strong performance gain. On the downside we mention the fact that GPUs implement single-precision floating point arithmetic, although this is beginning to change with the newest generation of GPUs from NVIDIA. Another consideration is that the zoom approach can only deliver linear convergence, as opposed to the quadratic convergence of Newton iteration for the regular case.

6 ACKNOWLEDGEMENTS

This work has been supported in part by NSC Grant NSC 97-2212-E-031-002, NSF CPATH CCF-0722210, DOE DE-FG52-06NA26290, and by a gift from Intel Corp. Several of the referees suggested using a multi-resolution GPU approach and eliminate the numerical iteration altogether.

REFERENCES

[1] Bouma, W.; Fudos, I.; Hoffmann, C.; Cai, J.; Paige, R.: A Geometric Constraint Solver, CAD 27, 1995, 487-501.
[2] Chiang, C.-S.: The Euclidean Distance Transform, PhD Thesis, Computer Science Dept., Purdue University, 1992.
[3] Chiang, C.-S.; Joan-Arinyo, R.: Revisiting Variable Radius Circles in Constructive Geometric Constraint Solving, CAD 21(4), 2004, 371-399.
[4] Chiang, C.-S.; Tsai, S.-H.; Chen, J.: The Control Vector Scheme for Design of Planar Primitive PH curves, International Journal of Mathematics Sciences, 1(4), 2007.
[5] Chiang, C.-S.; Lin, S.-Y.; Chen, J.C.: The Cyclographic Maps for Bezier Curve, 35th International Conference on Computers and Industrial Engineering, 2005, 453-458; Istanbul, Turkey.
[6] Chiang, C.-S.; Hoffmann, C.: Apollonius meets Pythagoras, manuscript, 2009.
[7] Durand, C. B.: Symbolic and numerical techniques for constraint solving, PhD thesis, Computer Science Department, Purdue University, 1998.
[8] Farin, G.: Curves and surfaces for computer aided geometric design: a practical guide, Academic Press, Boston, 1993.
[9] Farouki, R.; Sakkalis, T.: Pythagorean Hodographs, IBM J. Res. And Dev. 34(5), September 1990.
[10] Farouki, R.: Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable, Springer-Verlag, Berlin, Heidelberg, New York, 2008.
[11] Fudos, I.: Constraint-Based Parametric Conics for CAD, CAD 28, 1996, 91-100.
[12] Heo, H.-S.; Kim, M.-S.; Elber, G.: The intersection of two ruled surfaces, CAD 31, 1999, 33-50.
[13] Hoffmann, C.; Lomonosov, A.; Sitharam, M.; Decomposition plans for geometric constraint systems, Part I: Performance measures for CAD, Journal of Symbolic Computation, 31(4), 2001, 367–408.
[14] Hoffmann, C.; Chiang, C.-S.: Variable-Radius Circles in Cluster Merging Part I: Translational Cluster, CAD 34, 2002, 787-798.
[15] Hoffmann, C.; Chiang, C.-S.: Variable-Radius Circles in Cluster Merging Part II: Rotational Cluster, CAD 34, Sep., 2002, 799-806.
[16] Hoff III, K.; Culver, T.; Keyser, J.; Lin, M.; Manocha, D.: Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware, Siggraph '99, 277-286, Los Angeles, CA.
[17] Lamure, H.; Michelucci, D.: Solving geometric constraints by homotopy, Proc. Symp. on Solid Modeling and Applications, 1995, 263-269, ACM Press.
[18] Müller, E.; Krames, J.: Die Zyklographie, Franz Deuticke, Leipzig und Wien, 1929.
[19] Owen, J.: Algebraic solution for geometry from dimensional constraints, in SMA'91: Proceedings of the first ACM symposium on Solid modeling Foundations and CAD/CAM applications, 1991, 397–407, ACM Press, New York.
[20] Owens, J.; Luebke, D.; Govindaraju, N.; Harris, M.; Krüger, J.; Lefohn, A.; Purcell, T.: A Survey of General-Purpose Computation on Graphics Hardware. Eurographics 2005, State of the Art Reports, 2005, 21-51.
[21] Peternell, M.: Geometric properties of bisector surfaces, Graphical Models 62, 2000, 202-236.
[22] Sederberg, T.; Saito, T.: Rational-Ruled Surfaces: Implicitization and Section Curves, Graph. Models and Image Proc. 57, 1995, 334-342.
[23] Sud, A.; Govindaraju, N.; Gayle, R.; Manocha, D.: Interactive 3D Distance Field Computation using Linear Factorization, ACM Symp. on Interactive 3D Graphics and Games (I3D), 2006, 117-124.
[24] van der Meiden, H. A.: Semantics of Families of Objects, PhD Thesis, Delft University of Technology, The Netherlands, 2008.