Final version available at ScienceDirect : http://dx.doi.org/10.1016/j.cam.2014.03.013

Detecting Similarity of Rational Plane Curves

Juan Gerardo Alcázar[a,1], Carlos Hermoso[a], Georg Muntingh[b,2]

[a] *Departamento de Física y Matemáticas, Universidad de Alcalá, E-28871 Madrid, Spain*
[b] *SINTEF ICT, PO Box 124 Blindern, 0314 Oslo, Norway, and*
Department of Mathematics, University of Oslo, PO Box 1053, Blindern, 0316 Oslo, Norway

Abstract

A novel and deterministic algorithm is presented to detect whether two given rational plane curves are related by means of a similarity, which is a central question in Pattern Recognition. As a by-product it finds all such similarities, and the particular case of equal curves yields all symmetries. A complete theoretical description of the method is provided, and the method has been implemented and tested in the Sage system for curves of moderate degrees.

1. Introduction

A central problem in Pattern Recognition and Computer Vision is to detect whether a certain object corresponds to one of the objects stored in a database. The goal is to identify the given object as one of the objects in the database, and therefore to classify it or to consider it as unknown. When the objects involved in the recognition process are planar and defined by their silhouettes, algebraic curves can be used. For instance, in this setting silhouettes of aircraft prototypes [14] and of sea animals [30] have previously been considered.

However, the two objects to be compared need not be in the same position, orientation, or scale. In order to compare the two objects, it should therefore be checked whether there exists a nontrivial movement, also called *similarity*, transforming the object to be analyzed into the possible target in the database. This problem, known in the Computer Vision literature as *pose estimation*, has been extensively considered using many different techniques, using B-splines [14], Fourier descriptors [27], complex representations [30], statistics [12, 18, 22] also in the 3D case, moments [29, 31], geometric invariants [32, 35, 36], Newton-Puiseux parametrizations [23], and differential invariants [6, 7, 34]. The

Email addresses: `juange.alcazar@uah.es` (Juan Gerardo Alcázar), `carlos.hermoso@uah.es` (Carlos Hermoso), `georgmu@math.uio.no` (Georg Muntingh)

[1] Supported by the Spanish "Ministerio de Ciencia e Innovacion" under the Project MTM2011-25816-C02-01. Partially supported by a José Castillejos' grant from the Spanish Ministerio de Educación, Cultura y Deporte. Member of the Research Group ASYNACS (Ref. CCEE2011/R34)

[2] Partially supported by the Giner de los Ríos grant from the Universidad de Alcalá.

interested reader may consult the bibliographies in these papers to find other references on the matter.

With exception for the references concerning the B-splines and differential invariants, the above methods use the implicit form of the curves. Moreover, the above methods are either numerical, or only efficient when considered in a numerical setting. The reason for this is that in Pattern Recognition it is often assumed that the inputs are "fuzzy". For instance, in many cases the objects are represented discretely as point clouds. In this situation, an implicit equation is usually first computed for each cloud, and the comparison is performed later. In other cases there may be occluded parts or noise in the input. Alternatively the input might be exact, but modeling a real object only up to a certain extent. In all of these cases we do not need an perfect matching, so that a numerical comparison is sufficient.

In this paper, we address the problem from a perspective that differs in two ways. First of all, we assume that the curves are given in *exact* arithmetic, so that we can provide a deterministic answer to the question whether these two curves are similar. If so, our algorithm will find the similarities transforming one into the other. A second difference is that we start from rationally, or piecewise rationally, *parametrized* curves and carry out all computations in the parameter space. As a consequence, the cost of converting to the implicit form, both in terms of computing time and growth of the coefficients, is avoided. This representation is important in applications of Computer-Aided Geometric Design. In CAD-based systems, for instance, curves are typically represented by means of piecewise-rational parametrizations, usually of moderate degrees, like (rational) Bézier curves, splines and NURBS. Therefore, if the comparison is to be made between these representations, an algorithm based on the parametric form is desirable.

A first potential application of the ideas in the paper is related to computer algebra systems. Assume that a database with classical curves is stored in your favourite computer algebra system. Using the algorithms in this paper, the system can recognize a certain curve introduced by the user as one of the curves in the database. This way, the user can identify a curve as, say, a cardioid, an epitrochoid, a deltoid, etc.

A second application is related to Computer Graphics. In this field, the recognition of similarities simplifies manipulating and storing images, since in the presence of a similarity we need to store only one image, and the similarity producing the other. In the case of curves represented by Bézier curves or splines, one can detect similarity by checking whether the corresponding control polygons are similar. This is easy, because the control polygons are piecewise linear objects. However, this is less clear in the case of rational Bézier curves or NURBS. Furthermore, if one is not interested in global similarities but in *partial* similarities, i.e., in determining whether parts of the objects are similar, then it is no longer clear how to derive this from the control polygons. In addition, any algorithm to find the similarities between rational curves is also an algorithm to find the symmetries of a rational curve, since an algebraic curve is self-similar if and only if it is symmetric; see Proposition 2. Symmetry detection has been massively addressed in the field of Computer Graphics to gain understanding when analyzing pictures, and also in order to perform tasks like compression, shape editing, and shape completion; see for instance [4, 5, 19, 20, 21, 24], and the references therein.

We exploit and generalize some ideas used in [1, 2] for the computation of symmetries of rational plane curves, improving the algorithm provided in [2]. Our approach exploits the rational parametrizations to reduce to calculations in the parameter domain, and therefore to operations on univariate polynomials. Thus we proceed symbolically to determine the existence and computation of such similarities, using basic polynomial multiplication, GCD-computations. Additionally, if a numerical approximation is desired, univariate polynomial real-solving must be used as well. We have implemented and tested this algorithm in the Sage system [28]. It is also worth mentioning that, as a by-product, we achieve an algorithm to detect whether a given rational curve is symmetric and to find such symmetries. This problem has previously been studied from a deterministic point of view [1, 2, 8, 16], and by many authors from an approximate point of view.

The structure of the paper is the following. Some generalities on similarities and symmetries, to be used throughout the paper, are established in Section 2. The method itself is addressed in Section 3, first for polynomially parametrized curves and then for the general case. The case of piecewise rational curves is addressed at the end of this section. Finally, practical details on the implementation, including timings, are provided in Section 4.

2. Similarities and symmetries

Throughout the paper, we consider rational plane algebraic curves $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{R}^2$ that are neither a line nor a circle. Such curves are *irreducible*, i.e., they are the zero sets of polynomials that can not be factored over the reals, and can be parametrized by rational maps

$$\phi_j : \mathbb{R} \dashrightarrow C_j \subset \mathbb{R}^2, \qquad \phi_j(t) = \big(x_j(t), y_j(t)\big), \qquad j = 1, 2. \tag{1}$$

The components x_j, y_j of ϕ_j are rational functions of t, and they are defined for all but a finite number of values of t. We assume that the parametrizations (1) are *proper*, i.e., birational or, equivalently, injective except for perhaps finitely many values of t. In particular, the parametrizations have a rational inverse defined on their images. This is no restriction on the curves $\mathcal{C}_1, \mathcal{C}_2$, as any rational curve always admits a proper parametrization. For proofs of these claims and for a thorough study on properness, see [26, §4.2].

Roughly speaking, an *(affine) similarity* of the plane is a linear affine map from the plane to itself that preserves ratios of distances. More precisely, a map $f : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ is a similarity if $f(x) = Ax + b$ for an invertible matrix $A \in \mathbb{R}^{2\times 2}$ and a vector $b \in \mathbb{R}^2$, and there exists an $r > 0$ such that

$$\|f(x) - f(y)\|_2 = r\|x - y\|_2, \qquad x, y \in \mathbb{R}^2,$$

where $\| \cdot \|_2$ denotes the Euclidean norm. We refer to r as the *ratio* of the similarity. Notice that if $r = 1$ then f is an *(affine) isometry*, in the sense that it preserves distances. The similarities of the plane form a group under composition, and the isometries form a subgroup. It is well known, and easy to derive, that a similarity can be decomposed into a translation, an orthogonal transformation, and a uniform scaling by its ratio r. The curves $\mathcal{C}_1, \mathcal{C}_2$ are *similar*, if one is the image of the other under a similarity.

For analyzing the similarities of the plane, we have found it useful to identify the Euclidean plane with the complex plane. Through this correspondence $(x, y) \simeq x + iy$, the parametrizations (1) correspond to parametrizations

$$z_j : \mathbb{R} \dashrightarrow C_j \subset \mathbb{C}, \qquad z_j(t) = x_j(t) + iy_j(t), \qquad j = 1, 2. \tag{2}$$

We can distinguish two cases for a similarity of the complex plane. A similarity f is either *orientation preserving*, in which case it takes the form $f(z) = \boldsymbol{a}z + \boldsymbol{b}$, or *orientation reversing*, in which case it takes the form $f(z) = \boldsymbol{a}\overline{z} + \boldsymbol{b}$. In each case, its ratio $r = |\boldsymbol{a}|$.

A *Möbius transformation* is a rational function

$$\varphi : \mathbb{R} \dashrightarrow \mathbb{R}, \qquad \varphi(t) = \frac{\alpha t + \beta}{\gamma t + \delta}, \qquad \Delta := \alpha\delta - \beta\gamma \neq 0. \tag{3}$$

It is well known that the birational functions on the line are the Möbius transformations [26].

Theorem 1. *Let $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{C}$ be rational plane curves with proper parametrizations $z_1, z_2 : \mathbb{R} \dashrightarrow \mathbb{C}$. Then $\mathcal{C}_1$, $\mathcal{C}_2$ are similar if and only if there exists a similarity f and a Möbius transformation φ such that*

$$z_2\big(\varphi(t)\big) = f\big(z_1(t)\big). \tag{4}$$

Moreover, if $\mathcal{C}_1, \mathcal{C}_2$ are similar by a similarity f, then there exists a unique Möbius transformation φ satisfying (4)*.*

Equation (4) is equivalent to the existence of a commutative diagram

$$\begin{tikzcd} \mathbb{C} \arrow[r, "f"] & \mathbb{C} \\ \mathbb{R} \arrow[u, dashed, "z_1"] \arrow[r, dashed, "\varphi"'] & \mathbb{R} \arrow[u, dashed, "z_2"'] \end{tikzcd} \tag{5}$$

which relates the problem of finding a similarity "upstairs" to finding a corresponding Möbius transformation "downstairs".

Proof. Suppose that $\mathcal{C}_1$ and $\mathcal{C}_2$ are similar. Then there exists a similarity f of the plane that restricts to a bijection $f : \mathcal{C}_1 \longrightarrow \mathcal{C}_2$. Since z_1, z_2 are proper, their inverses z_1^{-1}, z_2^{-1} exist as rational functions on $\mathcal{C}_1, \mathcal{C}_2$. The composition $z_2^{-1} \circ f \circ z_1$ is therefore also a rational function with inverse $z_1^{-1} \circ f^{-1} \circ z_2$, which is rational as well. Hence, $\varphi = z_2^{-1} \circ f \circ z_1$ is a birational transformation on the real line, implying that it must be a Möbius transformation. Clearly this choice of f and φ makes the diagram (5) commute.

Conversely, suppose we are given a similarity f of the plane and a Möbius transformation φ that makes the diagram (5) commute. The similarity f maps any point $z \in \mathcal{C}_1$ to $f(z) = z_2 \circ \varphi \circ z_1^{-1}(z)$, which lies in the image of z_2 and therefore on the curve $\mathcal{C}_2$. It follows that $\mathcal{C}_1$ and $\mathcal{C}_2$ must be similar.

For the final claim, suppose that there are two Möbius transformations φ_1, φ_2 making the diagram (5) commute. Then $z_2\big(\varphi_1(t)\big) = f\big(z_1(t)\big) = z_2\big(\varphi_2(t)\big)$. Since the parametrization z_2 is proper, we conclude $\varphi_1 = \varphi_2$. □

Although the final claim refers to a Möbius transformation φ that is unique, its coefficients, i.e., $\alpha, \beta, \gamma, \delta$ in (3), are only defined up to a common multiple. Since φ maps the real line to itself, these coefficients can always be assumed to be real by dividing by a common complex number if necessary.

Depending on whether f preserves or reverses the orientation of the complex plane, the diagram (5) will either take the form

$$z_2\big(\varphi(t)\big) = \boldsymbol{a} z_1(t) + \boldsymbol{b}, \tag{6}$$

or

$$z_2\big(\varphi(t)\big) = \boldsymbol{a}\overline{z_1(t)} + \boldsymbol{b}. \tag{7}$$

Next, let $\mathcal{C}$ be a rational plane curve, neither a line nor a circle, given by a proper rational map

$$\phi : \mathbb{R} \dashrightarrow \mathcal{C} \subset \mathbb{R}^2, \qquad \phi(t) = \big(x(t), y(t)\big).$$

A *self-similarity* f of $\mathcal{C}$ is a similarity of the plane satisfying $f(\mathcal{C}) = \mathcal{C}$.

Proposition 2. *Let $\mathcal{C}$ be an algebraic curve that is not a union of (possibly complex) concurrent lines. Any self-similarity f of $\mathcal{C}$ is an isometry.*

Proof. Writing $z = x + iy$, the curve $\mathcal{C}$ is the zeroset of a polynomial

$$G(z, \bar{z}) = \sum_{k=0}^{d} G_k(z, \bar{z}),$$

where $G_k(z, \bar{z})$ is homogeneous of degree k and $G_d(z, \bar{z})$ is nonzero. Moreover, since $\mathcal{C}$ is not the union of concurrent lines, at least one other form $G_l(z, \bar{z})$, with $0 \leq l \leq d-1$, must be nonzero.

Let us first consider the case that $f(z) = \boldsymbol{a}z + \boldsymbol{b}$ is an orientation-preserving symmetry. Suppose that f is not an isometry, i.e., $|\boldsymbol{a}| \neq 1$. Then f has a unique fixed point $\boldsymbol{b}/(1 - \boldsymbol{a})$, which we can without loss of generality assume to be located at the origin by applying an affine change of coordinates if necessary. Hence, in these coordinates $\mathcal{C}$ has the symmetry $f(z) = \boldsymbol{a}z$, so that it is also the zeroset of $G(\boldsymbol{a}z, \bar{\boldsymbol{a}}\bar{z})$, which therefore must be a scalar multiple of $G(z, \bar{z})$. Since this scalar multiple must be the same for G_l and G_d, one has $\boldsymbol{a}^i \bar{\boldsymbol{a}}^{l-i} = \boldsymbol{a}^j \bar{\boldsymbol{a}}^{d-j}$

for some i and j, implying $|\boldsymbol{a}|^l = |\boldsymbol{a}|^d$ and therefore contradicting our assumption that $|\boldsymbol{a}| \neq 1$.

Next consider the case that $f(z) = \boldsymbol{a}\bar{z} + \boldsymbol{b}$ is an orientation-reversing symmetry. Then it also has the (possibly trivial) orientation-preserving symmetry $f\big(f(z)\big) = \boldsymbol{a}\bar{\boldsymbol{a}}z + \boldsymbol{b} + \boldsymbol{a}\bar{\boldsymbol{b}}$, which must be an isometry by the above paragraph. It follows that $|\boldsymbol{a}|^2 = 1$ and therefore that f must be an isometry. □

The isometries of the plane form a group under composition, which consists of *reflections* $f(z) = \mathrm{e}^{i\theta}\big(\overline{z - \boldsymbol{b}}\big) + \boldsymbol{b}$, which reflect in the line $\Im\big(\mathrm{e}^{-i\theta/2}z - \mathrm{e}^{-i\theta/2}\boldsymbol{b}\big) = 0$, *rotations* $f(z) = \mathrm{e}^{i\theta}(z - \boldsymbol{b}) + \boldsymbol{b}$, which rotate around a point $\boldsymbol{b}$, *translations* $f(z) = z + \boldsymbol{b}$, and *glide reflections*, which are a composition of a reflection and a translation [9].

An isometry of the plane leaving $\mathcal{C}$ invariant, is more commonly known as a *symmetry* of $\mathcal{C}$. When $\mathcal{C}$ is different from a line it cannot be invariant under a translation or a glide reflection, as this would imply the existence of a line intersecting the curve in infinitely many points, contradicting Bézout's theorem. The remaining symmetries are therefore the *mirror symmetries* (reflections) and the *rotation symmetries*. The special case of *central symmetries* is of particular interest and corresponds to rotation by $\theta = \pi$. Notice that the identity map is a symmetry of any curve $\mathcal{C}$, called the *trivial symmetry*.

Symmetries of algebraic curves and the similarities between them are related as follows.

Theorem 3. *Let $\mathcal{C}_1, \mathcal{C}_2$ be two distinct, irreducible, similar, algebraic curves, neither of them a line or a circle. Then there are at most finitely many similarities mapping $\mathcal{C}_1$ to $\mathcal{C}_2$. Furthermore, there is a unique similarity mapping $\mathcal{C}_1$ to $\mathcal{C}_2$ if and only if $\mathcal{C}_1$ (and therefore also $\mathcal{C}_2$) only has the trivial symmetry.*

Proof. If $f, \tilde{f}$ are two similarities of the plane mapping $\mathcal{C}_1$ to $\mathcal{C}_2$, then $g := \tilde{f}^{-1} \circ f$ is also a similarity, which maps $\mathcal{C}_1$ into itself. As $\mathcal{C}_1$ is irreducible and not a line, Proposition 2 implies that g is a symmetry of $\mathcal{C}_1$. Since an algebraic curve that is neither a line nor a circle has at most finitely many symmetries [17, §5], there can therefore be at most finitely many similarities mapping $\mathcal{C}_1$ to $\mathcal{C}_2$.

For the second claim, if $\mathcal{C}_1$ only has the trivial symmetry, $\tilde{f}^{-1} \circ f$ is the identity and $\tilde{f} = f$ is the unique similarity mapping $\mathcal{C}_1$ to $\mathcal{C}_2$. Conversely, if $\mathcal{C}_1$ has a nontrivial symmetry $\tilde{g}$, then $f \circ \tilde{g}$ is an additional similarity mapping $\mathcal{C}_1$ to $\mathcal{C}_2$. □

3. Detecting similarity between rational curves

In this section we derive a procedure for detecting whether the curves $\mathcal{C}_1, \mathcal{C}_2$ are similar by an orientation preserving similarity. In this case, by Theorem 1, Equation (6) holds for some similarity $f(z) = \boldsymbol{a}z + \boldsymbol{b}$ and a Möbius transformation φ. The case of an orientation reversing similarity is analogous, by replacing z_2 by $\overline{z_2}$.

We first attack the simpler case when z_1, z_2 are polynomial. After that we consider the case when either z_1 or z_2 is not polynomial, while distinguishing between $\delta \neq 0$ and $\delta = 0$, with δ as in (3). In each case our strategy is to eliminate $\boldsymbol{a}, \boldsymbol{b}$ and reduce to a simpler problem in the parameter space, i.e., downstairs in the diagram (5). Once we obtain the possible Möbius transformations downstairs, they can be lifted to corresponding similarities upstairs.

3.1. The polynomial case

A curve is *polynomial* if it admits a polynomial parametrization. It is easy to check whether a rational parametrization defines a polynomial curve, and to quickly compute a polynomial parametrization in that case; see [26] for both statements. As a consequence, we can assume polynomial curves to be polynomially parametrized, without loss of generality and without loss of significant computation power. Notice that if $\mathcal{C}_1, \mathcal{C}_2$ are similar and one of them is polynomial, then the other must be polynomial as well.

For polynomial parametrizations z_1, z_2, the right hand side in (4) is polynomial, implying that the left hand side is polynomial as well. It follows that in that case the corresponding Möbius transformation φ should be linear affine, i.e., $\varphi(t) = \alpha t + \beta$. In this section we assume that the curves $\mathcal{C}_1, \mathcal{C}_2$ are related by a similarity whose corresponding Möbius transformation is of this form. This includes similar polynomial curves as a special case, but also other rational curves needed for the case treated in Section 3.3.

We require some mild assumptions on our parametrization z_1, namely that z_1, and therefore also its derivatives z_1', z_1'', are well defined at $t = 0$ and that z_1', z_1'' are nonzero at $t = 0$. Notice that there are only finitely many parameters t for which one of these conditions does not hold, so these conditions hold after applying an appropriate, or even random, linear affine change of the parameter t.

Evaluating (6) at $t = 0$ yields

$$z_2(\beta) = \boldsymbol{a} z_1(0) + \boldsymbol{b}. \tag{8}$$

To get rid of $\boldsymbol{b}$, we differentiate (6) and evaluate at $t = 0$, which gives

$$z_2'(\beta)\alpha = \boldsymbol{a} z_1'(0). \tag{9}$$

Differentiating (6) twice and evaluating at $t = 0$ yields

$$z_2''(\beta)\alpha^2 = \boldsymbol{a} z_1''(0). \tag{10}$$

The unknowns $\boldsymbol{a}$ and α are nonzero since f and φ are invertible, and $z_2'(\beta), z_2''(\beta)$ are not identically zero because $\mathcal{C}_2$ is not a line. Dividing (10) by (9) yields

$$\alpha = \frac{z_1''(0)}{z_1'(0)} \cdot \frac{z_2'(\beta)}{z_2''(\beta)}, \tag{11}$$

which does not involve $\boldsymbol{a}$ or $\boldsymbol{b}$.

Since the Möbius transformation $\varphi(t) = \alpha t + \beta$ maps the real line to itself, both α and β are required to be real. That is, their imaginary parts $\Im(\alpha), \Im(\beta)$ are zero. The following lemma shows that this limits β to only finitely many candidates.

Lemma 4. *If $\Im(\alpha) = 0$ holds for infinitely many values of β, then $\mathcal{C}_2$ is a line.*

Proof. Writing $z_1''(0)/z_1'(0) = a + ib$ and $z_2(\beta) = x(\beta) + iy(\beta)$, the condition $\Im(\alpha) = 0$ is equivalent to

$$b\big(x'x'' + y'y''\big) + a\big(x''y' - x'y''\big) = 0.$$

If y' is identically 0, then $\mathcal{C}_2$ is a line and the result follows. So assume that y' is not identically 0. Then the above condition is equivalent to

$$b\frac{x'x'' + y'y''}{y'^2} + a\left(\frac{x'}{y'}\right)' = 0.$$

Changing to a new variable $z = x'/y'$ and dividing by $z^2 + 1$, we obtain

$$b\frac{zz'}{z^2+1} + a\frac{z'}{z^2+1} = -b\frac{y''}{y'}.$$

Integrating this equation yields

$$\frac{b}{2}\ln\left(x'^2 + y'^2\right) = -a\arctan\left(\frac{x'}{y'}\right) + k, \tag{12}$$

for some constant k. Since $z_1''(0)$ is nonzero, a, b cannot both be zero. If $b = 0$, then x'/y' is constant and $\mathcal{C}_2$ is a line. If $b \neq 0$, writing $x' + iy' = re^{i\theta}$, Equation (12) becomes $r = K\mathrm{e}^{-\theta a/b}$ for some nonzero constant K. If $a \neq 0$, this curve is a logarithmic spiral, which is a non-algebraic curve and therefore contradicts that x' and y' are rational. If $a = 0$, on the other hand, then the arc length $r = \sqrt{x'^2 + y'^2}$ of z_2 is constant, which again implies that $\mathcal{C}_2$ is a line [11]. □

As a consequence, we obtain a polynomial condition $\xi(\beta) = 0$ on β, where

$$\xi(\beta) := b\big(x'(\beta)x''(\beta) + y'(\beta)y''(\beta)\big) + a\big(x''(\beta)y'(\beta) - x'(\beta)y''(\beta)\big) \tag{13}$$

is the numerator of the imaginary part of α in (11), with a, b, x, y as above. Furthermore, by (11), any real zero β of ξ determines the Möbius transformation through

$$\alpha(\beta) = \Re\left(\frac{z_1''(0)}{z_1'(0)} \cdot \frac{z_2'(\beta)}{z_2''(\beta)}\right), \tag{14}$$

as long as β is not a zero or pole of z_2' or z_2''. Applying this to (9) and (8) yields

$$\boldsymbol{a}(\beta) = \frac{z_2'(\beta)}{z_1'(0)}\alpha(\beta), \qquad \boldsymbol{b}(\beta) = z_2(\beta) - \boldsymbol{a}(\beta)z_1(0), \tag{15}$$

determining the similarity corresponding to β.

Finally, coming back to (6), we can check whether there exists a real zero β of ξ, such that

$$z_2\big(\alpha(\beta)t + \beta\big) - \boldsymbol{a}(\beta)z_1(t) - \boldsymbol{b}(\beta) = 0 \tag{16}$$

for all t. This is a rational function $R(t) = P_1(t)/P_2(t)$, whose coefficients are rational functions of β. For (16) to hold for all t, the coefficients of the numerator $P_1(t)$, which are rational functions of β, have to be zero. Taking the GCD of the numerators of these coefficients and of $\xi(\beta)$ gives a polynomial $P(\beta)$. Finally let $Q(\beta)$ be the result of taking out from $P(\beta)$ all the factors that make the numerators and denominators of $\alpha(\beta), \boldsymbol{a}(\beta)$ and the denominator of $\boldsymbol{b}(\beta)$ vanish. Analogously, a polynomial $Q(\beta)$ can be found in the case of an orientation reversing similarity.

In the theorem below, let $Q(\beta)$ be derived as in this section under the assumption that the Möbius transformation φ is linear affine.

Theorem 5. *The curves $\mathcal{C}_1, \mathcal{C}_2$ are similar if and only if $Q(\beta)$ has at least one real root. More than that, in that case they are symmetric if and only if $Q(\beta)$ has several real roots.*

Proof. If $\mathcal{C}_1, \mathcal{C}_2$ are similar, then Theorem 1 implies that there exists a similarity f and a Möbius transformation φ for which either (6) or (7) holds. By construction, and since α, β can be taken real, this implies that $Q(\beta)$ has a real zero. Conversely, if β is a real zero of $Q(\beta)$, then $\alpha = \alpha(\beta)$ in (14) is also real, (14) – (15) are well defined, and $f(z) = \boldsymbol{a}(\beta)z + \boldsymbol{b}(\beta)$ is a similarity that satisfies (6) with $\varphi(t) = \alpha(\beta)t + \beta$. Theorem 1 then implies that $\mathcal{C}_1, \mathcal{C}_2$ are similar. The second statement follows from Theorem 3. □

Observe that $Q(\beta)$ cannot be identically zero: Indeed, in that case by Theorem 5 we would have infinitely many similarities between $\mathcal{C}_1$ and $\mathcal{C}_2$, therefore contradicting Theorem 3. The number of real roots of $Q(\beta)$ can be found using Sturm's Theorem [3, §2.2.2]. Additionally, there are many efficient algorithms for isolating the real roots of a polynomial, see for instance [3, Algorithm 10.4], that are implemented in most popular computer algebra systems.

Thus we obtain a recipe, spelled out as Algorithm `SimilarPol`, for detecting whether two rational curves are related by a similarity whose corresponding Möbius transformation is linear affine, which is the case for all similar polynomial curves.

Although we have not made it explicit in the above algorithm, one can compute the similarities mapping $\mathcal{C}_1$ to $\mathcal{C}_2$, either symbolically, i.e., in terms of a real root β of $Q(\beta)$, or numerically, at the additional cost of approximating the real roots of $Q(\beta)$ with any desired precision.

3.2. The rational case I: $\delta \neq 0$

Next assume that $\mathcal{C}_1, \mathcal{C}_2$ are not polynomial. In that case, we can no longer assume that the Möbius transformation φ in (3) is linear affine. We start considering the case when φ satisfies $\delta \neq 0$. After performing a linear affine change of the parameter t if necessary, we may and will assume that z_1 is well defined at $t = 0$ and that z_1', z_1'' are nonzero at $t = 0$. Furthermore, after dividing the coefficients of φ by δ, we can assume that $\delta = 1$ and that α, β, γ are real.

```
Algorithm SimilarPol
Require: Two proper rational parametrizations $z_1, z_2 : \mathbb{R} \dashrightarrow \mathbb{C}$ of curves $\mathcal{C}_1, \mathcal{C}_2$
    of equal degree greater than one, such that $z_1$ is well defined at $t = 0$ and
    $z_1', z_1''$ are nonzero at $t = 0$.
Ensure: Returns whether $\mathcal{C}_1, \mathcal{C}_2$ are related by a similarity that corresponds to
    a linear affine change of the parameter downstairs in (5).
 1: Look for orientation preserving similarities.
 2: Let $\xi(\beta)$ be the numerator of $\Im(\alpha)$, with $\alpha$ from (11).
 3: Find $\alpha(\beta)$ from (14).
 4: Find $\boldsymbol{a}(\beta)$ and $\boldsymbol{b}(\beta)$ from (15).
 5: Let $P(\beta)$ be the GCD of $\xi(\beta)$, and all the polynomial conditions obtained
    when substituting $\alpha(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ into (6).
 6: Let $Q(\beta)$ be the result of taking out from $P(\beta)$ any factor shared with a
    denominator of $\alpha(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ or a numerator of $\alpha(\beta), \boldsymbol{a}(\beta)$.
 7: If $Q(\beta)$ has real roots, return TRUE.
 8: Look for orientation reversing similarities.
 9: Replace $z_1 \leftarrow \bar{z}_1$ and proceed as in lines 1 – 7.
10: If the above step did not return TRUE, then return FALSE.
```

Evaluating (6) at $t = 0$ again gives (8), but differentiating (6) now yields

$$z_2'(\varphi(t)) \cdot \frac{\Delta}{(\gamma t + 1)^2} = \boldsymbol{a} z_1'(t). \tag{17}$$

Evaluating (17) at $t = 0$, we get

$$z_2'(\beta) \cdot \Delta = \boldsymbol{a} z_1'(0), \tag{18}$$

while differentiating (17) and evaluating at $t = 0$ gives

$$\Delta\big(\Delta z_2''(\beta) - 2\gamma z_2'(\beta)\big) = \boldsymbol{a} z_1''(0). \tag{19}$$

Dividing (19) by (18), using that $z_1'(0), \Delta \neq 0$, and solving for γ gives

$$\gamma = -\frac{z_1''(0)}{2z_1'(0)} + \Delta \frac{z_2''(\beta)}{2z_2'(\beta)}. \tag{20}$$

Assuming $\delta = 1$ forces γ (and also α, β) to be real. Writing

$$-\frac{z_1''(0)}{2z_1'(0)} = A + Bi, \qquad \frac{z_2''(\beta)}{2z_2'(\beta)} = C(\beta) + D(\beta)i,$$

and since Δ is also real, we have $B + \Delta \cdot D(\beta) = 0$, and therefore

$$\Delta(\beta) = -\frac{B}{D(\beta)}, \quad \gamma(\beta) = A + \Delta(\beta)C(\beta), \quad \alpha(\beta) = \Delta(\beta) + \beta\gamma(\beta). \tag{21}$$

The following lemma proves that the above expressions are well defined.

Lemma 6. *If* $\Im\left(\dfrac{z_2''(\beta)}{z_2'(\beta)}\right) = 0$ *holds for infinitely many* β*, then* $\mathcal{C}_2$ *is a line.*

Proof. Writing $z_2(\beta) = x(\beta) + iy(\beta)$, one observes

$$\frac{z_2''}{z_2'} = \frac{x'' + iy''}{x' + iy'} = \frac{x''x' + y'y''}{x'^2 + y'^2} + i\frac{x'y'' - x''y'}{x'^2 + y'^2}.$$

Therefore

$$\Im\left(\frac{z_2''}{z_2'}\right) = \frac{x'y'' - x''y'}{x'^2 + y'^2} = \frac{x'^2}{x'^2 + y'^2} \cdot \frac{x'y'' - x''y'}{x'^2} = \frac{x'^2}{x'^2 + y'^2} \cdot \frac{\mathrm{d}}{\mathrm{d}\beta}\left(\frac{y'}{x'}\right),$$

which is identically zero if and only if either x' is identical to zero or y'/x' is identical to a constant. In either case one deduces that $\mathcal{C}_2$ is a line. □

Hence the similarity corresponding to φ is found from (18) and (8), i.e.,

$$\boldsymbol{a}(\beta) = \frac{z_2'(\beta)}{z_1'(0)}\Delta(\beta), \qquad \boldsymbol{b}(\beta) = z_2(\beta) - \boldsymbol{a}(\beta)z_1(0). \tag{22}$$

Finally, substituting $\alpha(\beta), \gamma(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ into (6), and taking the GCD of the numerators as before, we again reach a polynomial condition $P(\beta) = 0$. Let $Q(\beta)$ be the polynomial obtained from $P(\beta)$ by taking out the factors that make a denominator of $\Delta(\beta), \alpha(\beta), \gamma(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ or a numerator of $\Delta(\beta), \boldsymbol{a}(\beta)$ vanish. We conclude that Theorem 5 also holds for similarities of this type, with $Q(\beta)$ defined under the assumption that the Möbius transformation φ has coefficient $\delta \neq 0$.

3.3. The rational case II: $\delta = 0$

The remaining case happens when $\mathcal{C}_1, \mathcal{C}_2$ are related by a similarity whose corresponding Möbius transformation (3) satisfies $\delta = 0$. Then $\gamma \neq 0$, and we may and will assume $\gamma = 1$ and therefore that α, β are real. Equation (6) becomes

$$z_2(\alpha + \beta/t) = \boldsymbol{a}z_1(t) + \boldsymbol{b}.$$

Making the change of parameter $t \longmapsto 1/t$, and defining $\tilde{z}_1(t) := z_1(1/t)$, we get

$$z_2(\alpha + \beta t) = \boldsymbol{a}\tilde{z}_1(t) + \boldsymbol{b}$$

It follows that $\mathcal{C}_1, \mathcal{C}_2$ are related by a similarity that corresponds to a linear affine change of parameters downstairs, which is the case of Section 3.1, with $\tilde{z}_1$ satisfying the same conditions as z_1. The similarities can then be detected by applying Algorithm `SimilarPol`.

We thus arrive at a recipe for detecting whether two rational curves are similar, spelled out as Algorithm `SimilarGen`. As in the case of `SimilarPol`, the corresponding similarities can be computed explicitly.

```
Algorithm SimilarGen
Require: Two proper rational parametrizations $z_1, z_2 : \mathbb{R} \dashrightarrow \mathbb{C}$ of curves $\mathcal{C}_1, \mathcal{C}_2$
    of equal degree, none of which is a line or a circle, such that $z_1(t), z_1(1/t)$
    are well defined at $t = 0$ and $z_1'(t), z_1''(t), \frac{\mathrm{d}}{\mathrm{d}t} z_1(1/t), \frac{\mathrm{d}^2}{\mathrm{d}t^2} z_1(1/t)$ are nonzero
    at $t = 0$.
Ensure: Returns whether $\mathcal{C}_1, \mathcal{C}_2$ are related by a similarity.
 1: Look for orientation preserving similarities with $\delta \neq 0$.
 2: Find $\Delta(\beta), \gamma(\beta), \alpha(\beta)$ from (21).
 3: Find $\boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ from (22).
 4: Let $P(\beta)$ be the GCD of all the polynomial conditions obtained when sub-
    stituting $\alpha(\beta), \gamma(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ in (6).
 5: Let $Q(\beta)$ the result of taking out from $P(\beta)$ any factor shared with a de-
    nominator of $\Delta(\beta), \alpha(\beta), \gamma(\beta), \boldsymbol{a}(\beta), \boldsymbol{b}(\beta)$ or a numerator of $\Delta(\beta), \boldsymbol{a}(\beta)$.
 6: If $Q(\beta)$ has a real root, return TRUE.
 7: Look for orientation reversing similarities with $\delta \neq 0$.
 8: Replace $z_1 \leftarrow \bar{z}_1$ and proceed as in lines $1 - 6$.
 9: Look for the remaining similarities with $\delta = 0$.
10: Replace $(\alpha, \beta) \leftarrow (\beta, \alpha)$ and $z_1(t) \leftarrow z_1(1/t)$.
11: Return the result of SimilarPol.
```

3.4. The similarity type

Notice that if two curves $\mathcal{C}_1, \mathcal{C}_2$ are identified as similar by Algorithm `SimilarPol` or `SimilarGen`, and if β is a real root of Q, then $\boldsymbol{a}(\beta)$ and $\boldsymbol{b}(\beta)$ define a similarity transforming $\mathcal{C}_1$ into $\mathcal{C}_2$. The nature of the similarity f can be found as follows by analyzing the fixed points.

In case f is orientation preserving, any fixed point z_0 satisfies $(1-\boldsymbol{a})z_0 = \boldsymbol{b}$. If $\boldsymbol{a} = 1$ and $\boldsymbol{b} = 0$ then every $z_0 \in \mathbb{C}$ is a fixed point, and f is the identity. If $\boldsymbol{a} = 1$ and $\boldsymbol{b} \neq 0$ then there are no fixed points, and f is a translation by $\boldsymbol{b}$. If $\boldsymbol{a} \neq 1$, then there is a unique fixed point $z_0 = \boldsymbol{b}/(1-\boldsymbol{a})$. Writing $\boldsymbol{a} = r\mathrm{e}^{i\theta}$, the similarity is a counter-clockwise rotation by an angle θ around the origin, followed by a scaling by r and a translation by $\boldsymbol{b}$.

In case f is orientation reversing, any fixed point z_0 satisfies $z_0 - \boldsymbol{a}\bar{z}_0 = \boldsymbol{b}$. If $\boldsymbol{a} = 1$ and $\boldsymbol{b} = 0$, then the x-axis is invariant and f is a reflection in the x-axis. If $\boldsymbol{a} = 1$ and $\boldsymbol{b} \neq 0$ is real, then the x-axis is invariant and f is a glide reflection, which first reflects in the x-axis and then translates by $\boldsymbol{b}$. If $\boldsymbol{a} = 1$ and $\Im(\boldsymbol{b}) \neq 0$, then there are no fixed points and f is a reflection in the x-axis, followed by a translation by $\boldsymbol{b}$. If $\boldsymbol{a} \neq 1$, write $\boldsymbol{a} = r\mathrm{e}^{i\theta}$. If $r = 1$ then there is a line of fixed points and f is an isometry that first reflects in the x-axis, then rotates counter-clockwise by θ, and then translates by $\boldsymbol{b}$. If $r \neq 1$ then there is a unique fixed point $z_0 = (\boldsymbol{a}\overline{\boldsymbol{b}} + \boldsymbol{b})/(1 - |\boldsymbol{a}|^2)$, and f first reflects in the x-axis, rotates counter-clockwise by θ around the origin, scales by r, and finally translates by b.

3.5. Piecewise rational curves

Let us now assume that $\mathcal{C}_1, \mathcal{C}_2$ are piecewise rational curves. This situation is common in Computer Aided Geometric Design, for instance, where objects are usually modeled by means of (rational) Bézier curves or NURBS, which are piecewise rational. In this setting, $\mathcal{C}_1, \mathcal{C}_2$ are parametrized by $\boldsymbol{p}_1(t), \boldsymbol{p}_2(t)$, where

$$\boldsymbol{p}_1(t) := \begin{cases} \boldsymbol{x}_1(t) & \text{if } t \in [t_1, t_2], \\ \boldsymbol{x}_2(t) & \text{if } t \in [t_2, t_3], \\ \quad\vdots & \qquad\vdots \\ \boldsymbol{x}_m(t) & \text{if } t \in [t_m, t_{m+1}], \end{cases} \qquad \boldsymbol{p}_2(t) := \begin{cases} \boldsymbol{y}_1(t) & \text{if } t \in [t_1, t_2], \\ \boldsymbol{y}_2(t) & \text{if } t \in [t_2, t_3], \\ \quad\vdots & \qquad\vdots \\ \boldsymbol{y}_n(t) & \text{if } t \in [t_n, t_{n+1}], \end{cases}$$

for some integers $m, n \geq 1$ and rational functions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ and $\boldsymbol{y}_1, \ldots \boldsymbol{y}_n$.

We want to detect whether $\mathcal{C}_1, \mathcal{C}_2$ are similar, or perhaps whether $\mathcal{C}_1, \mathcal{C}_2$ are *partially* similar, meaning that some part of $\mathcal{C}_1$ is similar to some part of $\mathcal{C}_2$. For this purpose we need to compare pieces of the curves, which are each described by a (rational) parametrization on a certain parameter interval, both of which need to be taken into account. Let $\boldsymbol{x}_i(I)$, $\boldsymbol{y}_j(J)$ be two such pieces on the intervals $I := [t_i, t_{i+1}]$, $J := [t_j, t_{j+1}]$. Then $\boldsymbol{x}_i(I)$ and $\boldsymbol{y}_j(J)$ are similar if and only if (1) the whole curves $\boldsymbol{x}_i(\mathbb{R})$ and $\boldsymbol{y}_j(\mathbb{R})$ are similar, and (2) the corresponding Möbius transformation φ satisfies that $\varphi(I) = J$. For $\mathcal{C}_1$ and $\mathcal{C}_2$ to be similar in this setting, every piece of $\mathcal{C}_1$ must be a similar with a piece of $\mathcal{C}_2$ and conversely, all with he same similarity. If (1) holds and (2) is replaced by the condition $J' := \varphi(I) \cap J \neq \emptyset$, then we have a partial similarity, in the sense that $\boldsymbol{y}_j\big(J'\big)$ is similar to $\boldsymbol{x}_i\big(\varphi^{-1}(J')\big)$.

3.6. Computing symmetries

From Proposition 2, we can find the symmetries of $\mathcal{C}$ by applying Algorithms `SimilarPol` and `SimilarGen` with both arguments the parametrization $z_1 = z_2$ of $\mathcal{C}$. Furthermore, the nature of the symmetry can be deduced from the set of fixed points, as observed in Section 3.4. Since an irreducible algebraic curve different from a line cannot have a nontrivial translation symmetry or glide-reflection symmetry, we are left with the cases mentioned in Section 2.

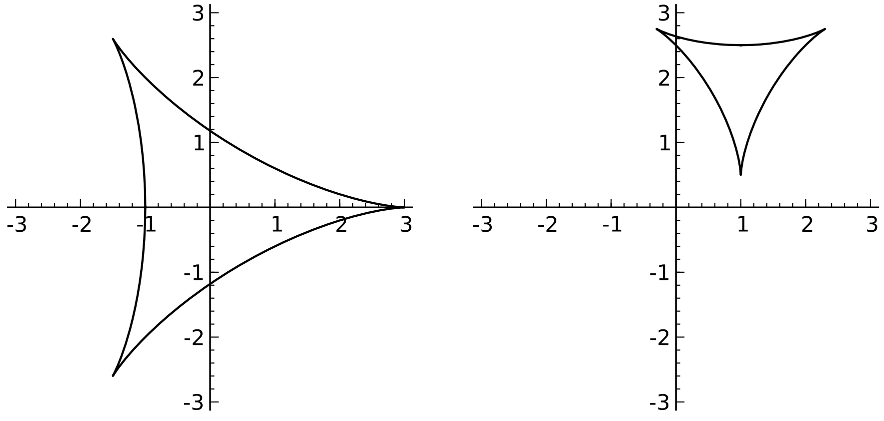
4. Implementation and experimentation

Algorithms `SimilarPol` and `SimilarGen` were implemented in Sage [28], using Singular [10] and FLINT [13] as a back-end. The corresponding worksheet "Detecting Similarity of Rational Plane Curves" is available online for viewing at the third author's website [25] and at https://cloud.sagemath.org, where it can be tried online once SageMath Cloud supports public worksheets.

4.1. An example: computing the symmetries of the deltoid

Let $\mathcal{C}_1 \subset \mathbb{R}^2$ be the *deltoid*, defined parametrically as the image of the rational map

$$\phi = (\phi_1, \phi_2) : \mathbb{R} \longrightarrow \mathbb{R}^2, \qquad t \longmapsto \left(\frac{-t^4 - 6t^2 + 3}{(t^2+1)^2}, \frac{8t^3}{(t^2+1)^2} \right), \tag{23}$$



(a)

(b)

Figure 1: The deltoid given as the image of (23) (left) and a related deltoid obtained by a similarity (right).

or implicitly as the zero locus of the equation

$$(x^2+y^2)^2 - 8(x^3-3xy^2) + 18(x^2+y^2) - 27 = 0.$$

As can be seen in Figure 1a, this curve is invariant under the symmetric group S_3 with six elements, which is generated by a 120°-rotation around the origin and the reflection in the horizontal axis.

Since $\phi'(0) = (0,0)$, this curve needs to be reparametrized before we can apply the algorithm. One checks that $z_1 = \phi_1(t-1) + \phi_2(t-1)i$ satisfies the conditions of the algorithm. To detect the symmetries of $\mathcal{C}_1$, we let $z_2 := z_1$ and apply the algorithm. We find $\Delta = \frac{1}{2}(\beta^2 - 2\beta + 2)$,

$$\gamma(\beta) = -\frac{\beta(2\beta^3 - 3\beta^2 - 18\beta + 22)}{4(\beta-1)(\beta^2-2\beta-2)},$$

$$\alpha(\beta) = -\frac{7\beta^4 - 34\beta^3 + 30\beta^2 + 8\beta - 8}{4(\beta-1)(\beta^2-2\beta-2)},$$

$$\boldsymbol{a}(\beta) = \frac{(\beta-1)(\beta^2-2\beta-2)}{(\beta^2-2\beta+2)^2}\big(2-\beta-\beta i\big), \tag{24}$$

$$\boldsymbol{b}(\beta) = 3\frac{\beta(\beta^2-6\beta+6)}{(\beta^2-2\beta+2)^2}\big(1-(\beta-1)i\big) \tag{25}$$

as functions of β. Substituting these expressions into

$$z_2\left(\frac{\alpha(\beta)t+\beta}{\gamma(\beta)t+1}\right) - \boldsymbol{a}(\beta)z_1(t) + \boldsymbol{b}(\beta)$$

yields a rational function in t whose coefficients are rational functions in β. Its numerator is a polynomial in t, and taking the GCD of the numerators of its coefficients, we find

$$\beta(\beta-1)(\beta^2-6\beta+6)(\beta^2-2\beta-2).$$

We are only interested in the β for which the expressions for $\alpha, \gamma, \Delta, \boldsymbol{a}$ and $\boldsymbol{b}$ are well-defined and for which the Möbius transformation and the similarity are invertible, so we discard the factors corresponding to the poles of $\alpha, \gamma, \Delta, \boldsymbol{a}, \boldsymbol{b}$ and zeros of $\Delta, \boldsymbol{a}$, leaving

$$\beta(\beta^2-6\beta+6).$$

Substituting the three zeros of this equation into Equations (24) and (25), we find three orientation preserving symmetries $f : z \longmapsto \boldsymbol{a}z + \boldsymbol{b}$, with $\boldsymbol{a} = 1, \mathrm{e}^{2\pi i/3}, \mathrm{e}^{4\pi i/3}$ and $\boldsymbol{b} = 0$, corresponding to rotations by 0°, 120°, and 240°. Repeating the procedure with $z_1 \mapsto \overline{z}_1$, we find the remaining three orientation reversing symmetries $f : z \longmapsto \boldsymbol{a}\overline{z} + \boldsymbol{b}$ in S_3 with again $\boldsymbol{b} = 0$ and $\boldsymbol{a} = 1, \mathrm{e}^{2\pi i/3}, \mathrm{e}^{4\pi i/3}$ and $\boldsymbol{b} = 0$.

4.2. An example: detecting similarities between two deltoids

Next, we let z_1 be as in Section 4.1 and define

$$z_2 : \mathbb{R} \longrightarrow \mathbb{C}, \qquad t \longmapsto \frac{t^4+4t^3+2t^2+1}{(t^2+1)^2} + \frac{5t^4+14t^2+1}{2(t^2+1)^2} i.$$

The corresponding curves $\mathcal{C}_1$ and $\mathcal{C}_2$ are shown in Figure 1. Applying the Algorithm `SimilarGen`, we obtain three orientation preserving similarities $f(z) = \boldsymbol{a}z + \boldsymbol{b}$ and three orientation reserving similarities $f(z) = \boldsymbol{a}\overline{z} + \boldsymbol{b}$, each with $\boldsymbol{b} = 1 + 2i$ and $\boldsymbol{a} \in \big\{ -\frac{1}{2}i, \frac{1}{2}\mathrm{e}^{\pi i/6}, \frac{1}{2}\mathrm{e}^{5\pi i/6} \big\}$. A direct computation yields

$$z_2(t-1) = -\frac{1}{2}i\big(\phi_1(t-1) + \phi_2(t-1)i\big) + 1 + 2i = -\frac{1}{2}iz_1(t) + 1 + 2i,$$

confirming that (6) holds with $\boldsymbol{a} = -i/2, \boldsymbol{b} = 1 + 2i$ and $\varphi(t) = t - 1$.

4.3. Observations on the complexity

For simplicity let us determine the complexity of Algorithm `SimilarPol`; the analysis of Algorithm `SimilarGen` is completely analogous. In addition to the standard Big O notation $\mathcal{O}$, we use the *Soft O* notation $\tilde{\mathcal{O}}$ to ignore any logarithmic factors in the complexity analysis.

Let z_1, z_2 be polynomials of degree at most d. Lines 1–4 can be carried out in $\mathcal{O}(d)$ (integer) operations, resulting in a polynomial ξ and rational functions $\alpha, \boldsymbol{a}$ and $\boldsymbol{b}$ whose numerators and denominators have degrees $\mathcal{O}(d)$. Observe that the bottleneck is line 5, where we substitute $\alpha(\beta)$ into Equation (6). Write

$$\alpha(\beta) = \Re\left(\frac{z_1''(0)}{z_1'(0)} \cdot \frac{z_2'(\beta)}{z_2''(\beta)}\right) =: \frac{\alpha_1(\beta)}{\alpha_2(\beta)},$$

for some relatively prime polynomials α_1, α_2 of degree at most $2d-3$. For (16) to hold identically for all t, we write $z_2(t) = c_d t^d + \cdots + c_1 t + c_0$ and compute

$$\sum_{l=0}^{d} P_l(\beta) t^l := \alpha_2^d(\beta)\Big[z_2\big(\alpha(\beta)t + \beta\big) - \boldsymbol{a}(\beta) z_1(t) - \boldsymbol{b}(\beta)\Big]$$

$$= \sum_{l=0}^{d} \left[\sum_{k=l}^{d} c_k \binom{l}{k} \alpha_2^{d-l}(\beta)\alpha_1^l(\beta)\beta^{k-l}\right] t^l - \alpha_2^d(\beta)\Big[\boldsymbol{a}(\beta) z_1(t) + \boldsymbol{b}(\beta)\Big].$$

Each polynomial P_l has degree $\mathcal{O}(d^2)$, and can be computed in $\tilde{\mathcal{O}}(d^3)$ operations by using binary exponentiation and FFT-based multiplication [33, §8.2]. The polynomial ξ from (13) has degree at most $2d-3$ and can be found in $\tilde{\mathcal{O}}(d)$ operations. Thus the polynomials $\xi, P_0, P_1, \ldots, P_d$ can be computed in $\tilde{\mathcal{O}}(d^4)$ operations. The GCD of two polynomials of degree at most d can be computed in $\tilde{\mathcal{O}}(d)$ operations [33, Corollary 11.6]. The GCD of the polynomials $\xi, P_0, P_1, \ldots, P_d$ can therefore be computed in $\tilde{\mathcal{O}}(d^3)$ operations and has degree $\mathcal{O}(d)$. At line 6 one takes the GCD Q of polynomials of degree $\mathcal{O}(d)$, which requires $\tilde{\mathcal{O}}(d)$ operations. To check, at the next line, whether Q has real roots can be done by root isolation, which takes $\tilde{\mathcal{O}}(d^3)$ operations [15, Theorem 17]. We conclude that the overall complexity is $\tilde{\mathcal{O}}(d^4)$. The analysis of Algorithm `SimilarGen` is cumbersome, but can be carried out similarly.

Of course the theoretical complexity is only a proxy for the practical performance of the algorithm, which also depends on the architecture, the precise implementation, and the bitsizes of the coefficients of the parametrizations.

4.4. Performance

Let the *bitsize* $\tau := \lceil \log_2 k \rceil + 1$ of an integer k be the number of bits needed to represent it. For various degrees d and bitsizes τ, we test the performance of Algorithms `SimilarPol` and `SimilarGen` by applying them to a pair of parametrized curves of equal degree d and bitsizes of their coefficients bounded by τ. For every pair (d, τ) this is done a number of times, and the average execution time t (CPU time) is listed in Tables 1 and 2 for a Dell XPS 15 laptop, with 2.4 GHz i5-2430M processor and 6 GB RAM.

Double logarithmic plots of the CPU times against the degrees and against the coefficient bitsizes are presented in Figures 2 and 3. In these plots the CPU times lie on a curve that seems to asymptotically approach a straight line, suggesting an underlying power law. The least-squares estimates of these asymptotes gives the power laws

$$t_{\text{pol}} = 2.17 \cdot 10^{-11} d^{9.28} \tau^{1.59}, \qquad t_{\text{gen}} = 6.2 \cdot 10^{-06} d^{9.91} \tau^{1.91} \tag{26}$$

for the CPU times t_{pol} of `SimilarPol` and t_{gen} of `SimilarGen`.

We should emphasize that these timings are for *dense* polynomials. The performance is much better in practice than Tables 1, 2 and Figures 2, 3 suggest.

To illustrate this, we find the symmetries of various well-known curves $z_1 : \mathbb{R} \longrightarrow \mathbb{C}$, namely the *folium of Descartes*,

$$t \longmapsto 3t\frac{1+ti}{1+t^3},$$

the *lemniscate of Bernoulli*,

$$t \longmapsto \frac{(3t^4+2t^3-2t-3)+(t^4+6t^3-6t-1)i}{5t^4+12t^3+30t^2+12t+5},$$

an *epitrochoid*,

$$t \longmapsto \frac{(-7t^4+288t^2+256)+(-80t^3+256t)i}{t^4+32t^2+256},$$

an *offset curve to a cardioid*,

$$t \longmapsto \frac{6t^8-756t^6+3456t^5-31104t^3+61236t^2-39366}{t^8+36t^6+486t^4+2916t^2+6561} \\ - \frac{18t(6t^6-16t^5-126t^4+864t^3-1134t^2-1296t+4374)}{t^8+36t^6+486t^4+2916t^2+6561}i,$$

a *hypocycloid* of degree 8,

$$t \longmapsto \frac{-3t^8-24t^7-120t^6-384t^5-680t^4-608t^3-224t^2+16}{t^8+8t^7+32t^6+80t^5+136t^4+160t^3+128t^2+64t+16} \\ + \frac{16t^7+112t^6+304t^5+400t^4+320t^3+256t^2+192t+64}{t^8+8t^7+32t^6+80t^5+136t^4+160t^3+128t^2+64t+16}i,$$

and *Rose curves*

$$t \longmapsto \frac{2t+(1-t^2)i}{(1+t^2)^{n+1}} \sum_{k=0}^{n} \binom{2n}{2k} (-1)^k t^{2k}, \qquad n = 2, 4, 6, 8, 10,$$

of degrees 6, 10, 14, 18, and 22. In each case the average CPU time used by `SimilarGen` is listed in Table 3. Observe that the degrees and coefficients of these examples are far from trivial. For more details we refer to the Sage worksheet [25].

To conclude, notice that the order of z_1 and z_2 is important for the performance of `SimilarGen`, so that one should let z_2 be the parametrization whose coefficients have the smallest bitsize.

t_{pol}	$\tau = 1$	$\tau = 8$	$\tau = 16$	$\tau = 32$	$\tau = 64$	$\tau = 128$
$d = 3$	0.024	0.022	0.024	0.020	0.030	0.038
$d = 6$	0.076	0.198	0.248	0.336	0.650	1.464
$d = 9$	0.764	2.316	3.220	6.626	15.151	42.753
$d = 12$	3.188	14.589	24.902	61.318	159.452	448.476
$d = 15$	11.885	90.340	206.469	587.249	1244.176	4439.909

Table 1: Average CPU time t_{pol} (seconds) of `SimilarPol` applied to random polynomial parametrizations of given degree d and with integer coefficients with bitsizes bounded by τ.

t_{gen}	$\tau = 1$	$\tau = 2$	$\tau = 4$	$\tau = 8$	$\tau = 16$	$\tau = 32$
$d = 2$	0.33	0.43	0.64	1.40	3.26	9.76
$d = 3$	1.50	3.90	8.53	21.94	85.00	387.69
$d = 4$	8.93	19.60	61.93	325.01	724.95	3881.44
$d = 5$	41.66	96.12	380.02	3956.86	22608.96	
$d = 6$	99.05	452.70	1028.22	17606.67		
$d = 7$	250.39	2670.24	19478.14			
$d = 8$	1805.01	21610.26				

Table 2: Average CPU time t_{gen} (seconds) of `SimilarGen` applied to random rational parametrizations of given degree d and with integer coefficients with bitsizes bounded by τ.

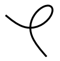
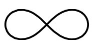
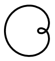
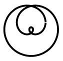
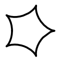
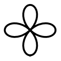
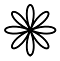
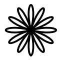
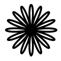
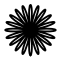
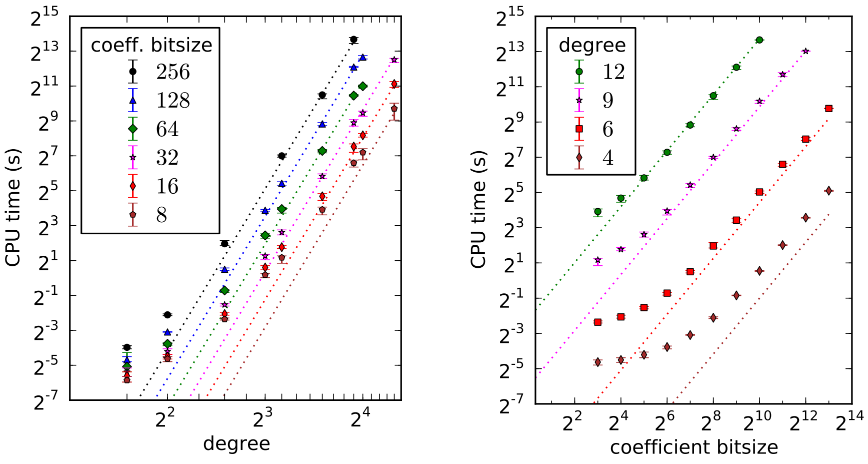
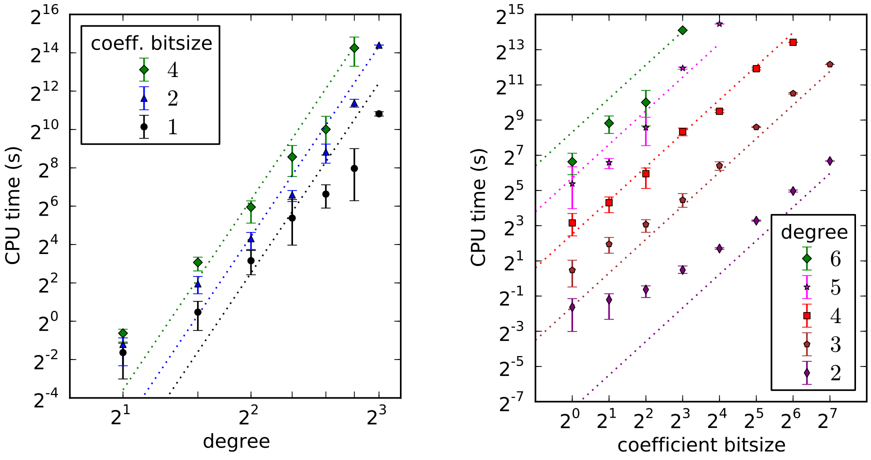
curve	Descartes' folium	Bernoulli's lemniscate	epitrochoid	cardioid offset	hypocycloid
					
degree	3	4	4	8	8
CPU time	0.11	0.32	0.11	0.77	3.59
curve	4-leaf rose	8-leaf rose	12-leaf rose	16-leaf rose	20-leaf rose
					
degree	6	10	14	18	22
CPU time	0.24	3.50	24.83	118.74	703.12

Table 3: Average CPU time (seconds) of `SimilarGen` for well-known curves.



(a) (b)

Figure 2: Double logarithmic plots of the average CPU time of `SimilarPol` versus degree (left) and versus the bitsize of the coefficients (right). The error bars show the range of CPU times found for the various random polynomials. The dotted line represents the fitted power law for t_{pol} in (26).



(a) (b)

Figure 3: Double logarithmic plots of the average CPU time of `SimilarGen` versus degree (left) and versus the bitsize of the coefficients (right). The error bars show the range of CPU times found for the various random polynomials. The dotted line represents the fitted power law for t_{gen} in (26).

References

[1] Alcazár J.G. (2014), *Efficient Detection of Symmetries of Polynomially Parametrized Curves*. Journal of Computational and Applied Mathematics vol. 255, pp. 715–724.

[2] Alcazár J.G., Hermoso C., Muntingh G. (2013), *Detecting Symmetries of Rational Plane Curves*. Submitted preprint available at `http://arxiv.org/abs/1207.4047`.

[3] Basu S., Pollack R., Roy M.F. (2006), *Algorithms in Real Algebraic Geometry*, Springer.

[4] Berner A., Bokeloh M., Wand M., Schilling A., Seidel H.P. (2008), *A Graph-Based Approach to Symmetry Detection*. Symposium on Volume and Point-Based Graphics (2008), pp. 1–8.

[5] Bokeloh M., Berner A., Wand M., Seidel H.P., Schilling A. (2009), *Symmetry Detection Using Line Features*. Computer Graphics Forum, Vol. 28, No. 2. (2009), pp. 697–706.

[6] Boutin M. (2000), *Numerically Invariant Signature Curves*, International Journal of Computer Vision 40(3), pp. 235–248.

[7] Calabi E., Olver P.J., Shakiban C., Tannenbaum A., Haker S. (1998), *Differential and Numerically Invariant Signature Curves Applied to Object Recognition*, International Journal of Computer Vision, 26(2), pp. 107–135.

[8] Carmichael, R. D. (1910), *On r-Fold Symmetry of Plane Algebraic Curves*, Amer. Math. Monthly 17, no. 3, pp. 56–64.

[9] Coxeter, H. S. M. (1969), *Introduction to geometry*, Second Edition, John Wiley & Sons, Inc., New York-London-Sydney.

[10] Decker W., Greuel G.-M., Pfister G., Schönemann H. (2011), Singular 3-1-3 — A computer algebra system for polynomial computations. `http://www.singular.uni-kl.de`.

[11] Farouki, Rida T., Sakkalis, Takis (1991), *Real rational curves are not "unit speed"*, Comput. Aided Geom. Design, 8(2), pp. 151–157.

[12] Gal R., Cohen-Or D. (2006), *Salient geometric features for partial shape matching and similarity*. ACM Transactions on Graphics, vol. 25(1), pp. 130–150.

[13] Hart, W.B. (2010), *Fast Library for Number Theory: An Introduction*, Mathematical Software – ICMS 2010, pp. 88–91, Lecture Notes in Comput. Sci., 6327, Springer, Berlin.

[14] Huang Z., Cohen F.S. (1996), *Affine-Invariant B-Spline Moments for Curve Matching*, IEEE Transactions on Image Processing, vol. 5, No. 10, pp. 1473–1480.

[15] Kerber M., Sagraloff M. (2012), *A worst-case bound for topology computation of algebraic curves*, Journal of Symbolic Computation, vol. 47, Issue 3, pp. 239–258.

[16] Lebmeir P., Richter-Gebert J. (2008), *Rotations, Translations and Symmetry Detection for Complexified Curves*, Computer Aided Geometric Design 25, pp. 707–719.

[17] Lebmeir P. (2009), *Feature Detection for Real Plane Algebraic Curves*, Ph.D. Thesis, Technische Universität München.

[18] Lei Z., Tasdizen T., Cooper D.B. (1998), *PIMs and Invariant Parts for Shape Recognition*, Proceedings Sixth International Conference on Computer Vision, pp. 827–832.

[19] Li M., Langbein F., Martin R. (2008), *Detecting approximate symmetries of discrete point subsets.* Computer-Aided Design vol. 40(1), pp. 76–93.

[20] Li M., Langbein F., Martin R. (2010), *Detecting design intent in approximate CAD models using symmetry.* Computer-Aided Design vol. 42(3), pp. 183–201.

[21] Mitra N.J., Guibas L.J., Pauly M. (2006), *Partial and approximate symmetry detection for 3d geometry*, ACM Transactions on Graph. 25 (3), pp. 560–568.

[22] Mishra R., Pradeep K. (2012), *Clustering Web Logs Using Similarity Upper Approximation with Different Similarity Measures*, International Journal of Machine Learning and Computing vol. 2, no. 3, pp. 219–221.

[23] Mozo-Fernández J., Munuera C. (2002), *Recognition of Polynomial Plane Curves Under Affine Transformations*, Applicable Algebra in Engineering, Communications and Computing, vol. 13, pp. 121–136.

[24] Podolak J., Shilane P., Golovinskiy A., Rusinkiewicz S., Funkhouser T. (2006), *A Planar-Reflective Symmetry Transform for 3D Shapes.* Proceeding SIGGRAPH 2006, pp. 549–559.

[25] Muntingh G., personal website, software
`https://sites.google.com/site/georgmuntingh/academics/software`

[26] Sendra J.R., Winkler F., Perez-Diaz S. (2008), *Rational Algebraic Curves*, Springer-Verlag.

[27] Sener S., Unel M. (2005), *Affine invariant fitting of algebraic curves using Fourier descriptors*, Pattern Analysis and Applications vol. 8, pp. 72–83.

[28] Stein, W. A. et al. (2013), *Sage Mathematics Software (Version 5.9)*, The Sage Development Team, `http://www.sagemath.org`.

[29] Suk T., Flusser J. (1993), *Pattern Recognition by Affine Moment Invariants*, Pattern Recognition, vol. 26, No. 1, pp. 167–174.

[30] Tarel J.P., Cooper D.B. (2000), *The Complex Representation of Algebraic Curves and Its Simple Exploitation for Pose Estimation and Invariant Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No. 7, pp. 663–674.

[31] Taubin G., Cooper D.B. (1992), *Object Recognition Based on Moments (or Algebraic) Invariants*, Geometric Invariance in Computer Vision, J.L. Mundy and A.Zisserman, eds., MIT Press, pp. 375–397, 1992.

[32] Unel M., Wolowich W.A. (2000), *On the Construction of Complete Sets of Geometric Invariants for Algebraic Curves*, Advances in Applied Mathematics, vol. 24, pp. 65–87.

[33] Von zur Gathen J., Gerhard, J. (2003) *Modern computer algebra. Second edition.* Cambridge University Press, Cambridge.

[34] Weiss I. (1993), *Noise-Resistant Invariants of Curves*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, No. 9, pp. 943–948.

[35] Wolowich W., Unel M. (1998), *The determination of implicit polynomial canonical curves*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20(10), pp. 1080–1089.

[36] Wolowich W., Unel M. (1998), *Vision-Based System Identification and State Estimation.* In: "The Confluence of Vision and Control, Lecture Notes in Control and Information Systems", New York, Springer-Verlag.