EUROGRAPHICS 2020 / U. Assarsson and D. Panozzo
(Guest Editors)

Volume 39 (2020), Number 2

Computational Design and Optimization of Non-Circular Gears

Hao Xu† [1] Tianwen Fu† [1] Peng Song[2] Mingjun Zhou[1] Chi-Wing Fu[1] Niloy J. Mitra[3,4]

[1]The Chinese University of Hong Kong [2]Singapore University of Technology and Design [3]University College London [4]Adobe Research London

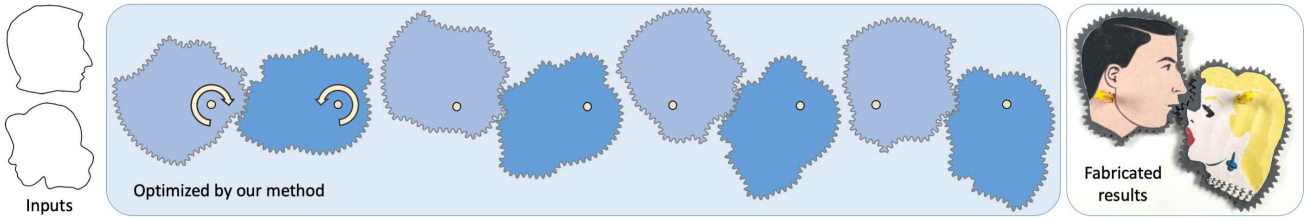


Figure 1: *We introduce an automatic method to design non-circular gears, which are optimized not only to resemble the input shapes (left) but also to transfer motion continuously and smoothly (middle). Further, our results can be 3D-printed and put to work in practice (right).*

Abstract

We study a general form of gears known as non-circular gears that can transfer periodic motion with variable speed through their irregular shapes and eccentric rotation centers. To design functional non-circular gears is nontrivial, since the gear pair must have compatible shape to keep in contact during motion, so the driver gear can push the follower to rotate via a bounded torque that the motor can exert. To address the challenge, we model the geometry, kinematics, and dynamics of non-circular gears, formulate the design problem as a shape optimization, and identify necessary independent variables in the optimization search. Taking a pair of 2D shapes as inputs, our method optimizes them into gears by locating the rotation center on each shape, minimally modifying each shape to form the gear's boundary, and constructing appropriate teeth for gear meshing. Our optimized gears not only resemble the inputs but can also drive the motion with relatively small torque. We demonstrate our method's usability by generating a rich variety of non-circular gears from various inputs and 3D printing several of them.

CCS Concepts

• ***Computing methodologies*** → *Shape modeling;* • ***Fabrication*** → *Computational design;*

1. Introduction

Gears are basic building blocks in mechanical devices. The most common form of gears is the circular gear, which transfers constant speed and torque from one gear to another. In this paper, we explore a general form of gears called *non-circular gears* [LFA04]. These gears have *non-circular* shapes and *eccentric* rotation centers, meaning that their boundaries are irregular and rotation centers are not necessarily at the gear centroid. In practice, non-circular gears can transfer periodic motion with variable speed, and are used in a variety of mechanical devices such as clockworks, astronomical devices, and musical instruments; see Figure 2 for examples.

Early works on non-circular gears focus on studying specific forms such as elliptical gears [Bai02] and square gears [KB99], analytically modeling their kinematic and dynamic behaviors, and empirically investigating their mechanical properties. Recently, the design of non-circular gears has received a lot of interest in the mechanical engineering community [LFAGPH09]. Some of the designed gears have been put to work in various mechanisms [CTN*13, ZHH*16]. However, the dominant design approach for non-circular gears is analytical and relies on the specification of a transmission function as the input. Hence, the approach is limited to professional users and impractical for designing gears of arbitrary shape; see Section 2 for details.

Motivated by the recent advancement of digital fabrication techniques, we aim to develop computational methods that enable general users to design *functional* non-circular gears with desired shape. This is a nontrivial problem for three reasons. First, unlike circular gears that can always couple together as long as their teeth have same spacing, a pair of non-circular gears should have compatible shape and gear profile to ensure almost unceasing contacts during the rotation; see Figure 1(middle). Second, the required instantaneous torque to drive non-circular gears is not constant but varies with distance (i.e., arm of force) between the contact

† joint first authors

Figure 2: *Non-circular gears designed by human experts: (a) eccentric gears sketched by Leonardo da Vinchi in "Codex Madrid II" [Lac08]; (b) elliptical gears by "Magic Wheel" [Whe07]; (c) wooden clocks with non-circular gears by Brian Law [Law13].*

and each rotation center. Hence, improperly-designed non-circular gears require extensive torque to drive, thereby reducing the mechanism's efficiency. Third, non-circular gears require teeth with customized shape to ensure proper gear meshing for transferring motion continuously without losing contact.

We approach the problem by first studying a pair of non-circular gears, modeling their geometry, kinematics, and dynamics, and also formulating necessary geometric and kinematic constraints to ensure a mating gear pair. Based on this foundation, we pose the problem of designing a pair of functional non-circular gears as a geometric optimization, taking a pair of user-specified 2D shapes as inputs. Our optimization has two goals: (i) the gears should resemble the input shapes; and (ii) the maximum instantaneous torque required to drive the gears should be minimized, with the constraint that the two gears should form a mating pair.

To solve the optimization, we minimally modify the input shapes into a pair of mating gears, while minimizing the required torque to drive them. Our idea is to construct a transmission function for each input shape with a fixed rotation center, then match and interpolate the transmission function derivatives to obtain the target transmission function. By then, we can construct the gear boundary shapes from the target transmission function, making the gears similar to the input shapes, while avoiding tedious and explicit shape deformations. To this end, we further devise a coarse-to-fine search to find optimal rotation centers that satisfy our two design objectives. Lastly, we create customized teeth on the resulting boundary shapes for gear meshing. Our method enables the creation of a rich variety of non-circular gears from various inputs. The results are demonstrated via simulations and 3D printing, e.g., see Figure 1.

2. Related Works

Non-circular gears. The idea of non-circular gears can be traced back to the 16th century, in a sketch by Leonardo da Vinci in "Codex Madrid II"; see Figure 2(a). Later, a number of intriguing mechanical properties of non-circular gears were discovered and studied in experiments and practice, such as variable speed transmission ratio [FLC00], force/moment balancing [AB10], and preventing resonance oscillations [KNF*17]. Non-circular gears have also been used in a variety of mechanisms, including planetary gear train [Mun06], geared linkage mechanism [MLB*09, CTN*13], and indexing mechanism [ZHH*16], for different applications.

Designing non-circular gears is nontrivial. Researchers in mechanical engineering typically use analytical methods to compute their geometry from a given transmission function [LFAGPH09, Bäs19]. However, specifying a transmission function requires deep, relevant knowledge in mathematics and engineering, which is beyond the capability of general users. Very recently, Valle designed a graphical software tool called Gearify [Val19], that can aid users to design customized non-circular gears. This software takes as input a user-provided gear profile, and automatically computes a mating gear that can mesh with the input gear. However, there is no control over the shape of the generated mating gear, making it difficult and tedious to design a pair of gears with desired shapes. Enabled by our computational method, our design tool takes two 2D shapes as input, and resorts to a geometric optimization to automatically find a pair of mating gears that resemble the input shapes.

Mechanism design. Computational design of personalized mechanical automata has excited great interest in the computer graphics community. The key challenge in this line of works is to design a mechanism that can effectively transfer an input motion (usually from a single rotational motor) to realize the user-specified motions or poses of the automata's end-effectors, while simultaneously satisfying the various fabrication, assembly, and aesthetic constraints.

To address the challenges, Zhang et al. [ZAC*17] retargeted an existing mechanism to a user-specified input shape, while Hergel and Lefebvre [HL15] generated mechanisms from 2D designs crafted by users. Rather than relying on existing mechanisms or user sketches, a number of research works gain more flexibility in designing the mechanisms by automatically constructing and connecting various classes of mechanical components, including traditional mechanical elements (such as cams [ZXS*12], gears [CTN*13, RCLM18], and linkages [TCG*14, BCT15, NBA19]), mechanical modules for oscillation [CLM*13], elemental mechanisms with higher pair joints [SWT*17], and even flexible mechanisms with compliant joints [MZB*17] or kinetic wires [XKCB18]. Among the above works, Coros et al. [CTN*13] incorporate non-circular gears for designing mechanical characters that can display user-specified motions, in which the relation between the gears profile and transmission function is also modelled. Compared with it, our work differs in two aspects. First, the design goals are different. We design non-circular gears to resemble the user-specified shapes, while they aim at controlling the timing of the motions. Second, the ways to generate the transmission function are different. We automatically generate the transmission function based on an optimization, while they let users specify it.

Non-circular gears designed by our method complement the mechanical components explored in existing works by enabling timing and force controls while transferring motion. This result could have great potential for use in the design of mechanisms that involve adaptive speed and force.

2D shape-guided design. Our work is also related to techniques that take 2D shapes as inputs to guide the design of an artwork, such that the artwork's contour resembles the given shape's boundary. Typical examples include calligram [ZCR*16], collage [KSH*16], and ornamental packing [SKAM17]. Other works take a pair of 2D shapes as input for designing dissection puzzles, including lattice-based dissection [ZW12], approximate dissection [DYYT17], and reversible hinged dissection [LMAH*18]. Similar to our work, the input shapes in [DYYT17, LMAH*18] are allowed to be slightly deformed to gain more flexibility when forming the designs.

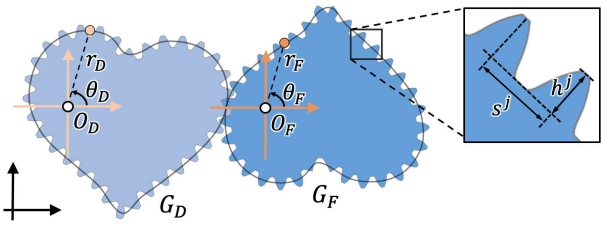


Figure 3: *Geometric parameters of a non-circular gear pair. As a convention, we color the driver (G_D) and follower (G_F) in light blue and dark blue, respectively, throughout the paper. The local coordinate system of G_D (G_F) is colored in light (dark) orange, and the world coordinate system is colored in black.*

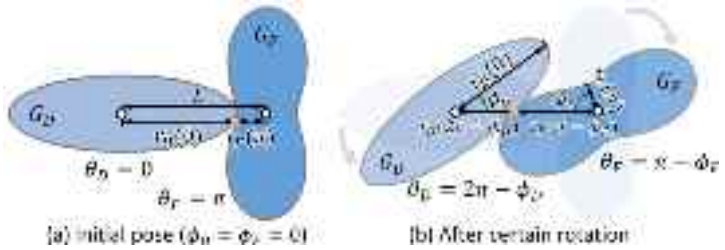


Figure 4: *A mating gear pair without teeth for better visualization. The contact point (orange dot) is always on the line between the two rotation centers.*

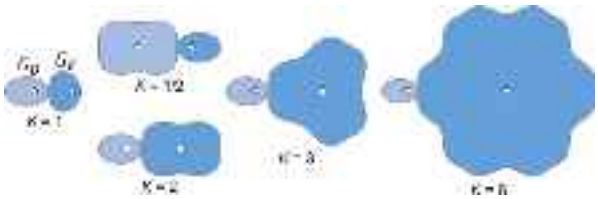


Figure 5: *Example non-circular gear pairs with different transmission ratio K. Teeth are omitted for better boundary visualization.*

3. Parameterizing Non-Circular Gears

3.1. Modeling Individual Non-Circular Gears

A non-circular gear has three geometric components: (i) a *closed boundary curve*; (ii) a *rotation center*; and (iii) *teeth* along the boundary. Following [LFAGPH09], we represent the *boundary curve* using a polar function $r(\theta)$, with rotation center at the origin of the local coordinate and r denoting the distance from the origin to curve at $\theta \in [0, 2\pi)$. In our implementation, we discretize $r(\theta)$ as a set of polar coordinates $\{\theta^i, r^i\}$, where $\theta^i = \frac{2\pi(i-1)}{N}$, $r^i = r(\theta^i)$, and $i \in \{1...N\}$. In practice, we set $N = 1024$ in all experiments.

Also, given M as the number of teeth on a gear, we use standard involute *tooth profile* [LFA04] to model the teeth using parameters $\{h^j\}$ and $\{s^j\}$: h^j is the height of the j-th tooth (from tooth root to tooth tip), s^j is the spacing to next tooth (see inset in Figure 3), $j \in \{1...M\}$, and j is uniformly sampled along boundary. The teeth geometry is constructed by adjusting the sampled points along the boundary curve to the locations computed based on $\{h^j\}$ and $\{s^j\}$.

3.2. Modeling Pairs of Mating Non-Circular Gears

Next, we model the geometry, kinematics, and dynamics of a non-circular gear pair, as well as the associated constraints.

Geometric modeling. In a pair of non-circular gears, the *driver* gear (denoted as G_D) initializes the motion and the *follower* gear (denoted as G_F) reacts and rotates accordingly. Their polar function and parameters are denoted as $\{\ r_D(\theta_D),\ \{\theta_D^i, r_D^i\},\ O_D,\ \{h_D^j, s_D^j\},\ M_D\ \}$, and $\{\ r_F(\theta_F),\ \{\theta_F^i, r_F^i\},\ O_F,\ \{h_F^k, s_F^k\},\ M_F\ \}$, respectively, where O_D and O_F represent the rotation centers of G_D and G_F, respectively, in the world coordinate system; see Figure 3.

The geometric parameters of the two gears are not independent. First, the perimeter of G_D should be K times the perimeter of G_F, in which K is a rational number (ratio of two positive integers), to enable gear meshing during periodic motion. Such perimeter ratio is in fact the *transmission ratio* between G_D and G_F. Note that for circular gears, K can be an arbitrary positive number as long as the mating gears have the same tooth spacing. This paper considers a specific class of non-circular gears, where K must be a positive integer, or a reciprocal of a positive integer, to simplify our design problem. Figure 5 showcases non-circular gear pairs with different K. When $K > 1$, the bigger gear is K-fold rotational symmetric, since it needs repetitive boundary and teeth profiles for compatible gear meshing. Lastly, we also have $M_F = K \times M_D$.

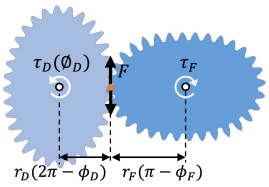
Kinematic modeling. The pose of G_F relative to G_D is *not* arbitrary. We must place G_F at a certain location relative to G_D as its initial pose. Without loss of generality, we define their initial poses as the ones shown in Figure 4(a), i.e., the two gears contact at $r_D(0)$ in G_D's coordinate system (with $\theta_D = 0$) and $r_F(\pi)$ in G_F's coordinate system (with $\theta_F = \pi$). Also, we denote L as the distance between the two rotation centers, so $L = r_D(0) + r_F(\pi)$.

Further, we denote ϕ_D and ϕ_F as the rotation angles of G_D and G_F, respectively, relative to initial poses, and set G_D to rotate counter-clockwise and G_F to rotate clockwise; see Figure 4(b). Then, we can define the *transmission function* for the gears:

$$\psi(\phi_D) = \phi_F\ , \tag{1}$$

which maps ϕ_D to ϕ_F; see again Figure 4(b). At the initial poses, $\psi(0) = 0$. Also, ψ is monotonic and periodic, since G_F keeps rotating in the same direction and both ϕ_D and ϕ_F vary periodically.

Dynamic modeling. We assume a constant torque (magnitude denoted as τ_F) that resists the follower gear to rotate, e.g., caused by friction or load. Dynamic modeling aims to compute the required instantaneous torque on the driver to move the gears. Such torque varies with ϕ_D, so its magnitude is denoted as $\tau_D(\phi_D)$. By the law of action and reaction, the constant force magnitude (denoted as F) acted on G_D and G_F should be the same; see the inset. So, by the formula of torque calculation, $\tau_D(\phi_D) = r_D(2\pi - \phi_D)F$ and $\tau_F = r_F(\pi - \phi_F)F$. Combining the two equations, we have



$$F = \frac{\tau_D(\phi_D)}{r_D(2\pi - \phi_D)} = \frac{\tau_F}{r_F(\pi - \phi_F)}\ . \tag{2}$$

So, the instantaneous torque required on the driver at ϕ_D is

$$\tau_D(\phi_D) = \frac{r_D(2\pi - \phi_D)}{r_F(\pi - \phi_F)}\tau_F\ . \tag{3}$$

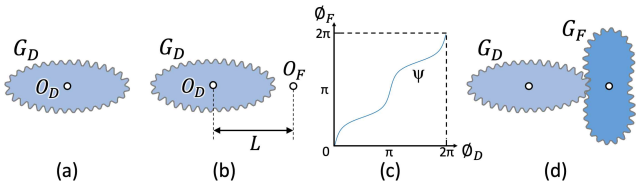


Figure 6: *Our approach to compute the mating gear. Given (a) G_D and $K = 1$, we first locate (b) G_F's rotation center O_F, then compute (c) the transmission function and (d) G_F's geometry.*

Note that we ignore energy loss, e.g., due to friction, in our model.

Geometric and kinematic constraints. Below, we summarize the constraints for ensuring smooth and continuous gear motion:

- First, since the distance between rotation centers of G_D and G_F is fixed, for any ϕ_D, we should have

$$r_D(2\pi-\phi_D) \;+\; r_F(\pi-\phi_F) \;=\; L\;,\;\text{where}\;\phi_F=\psi(\phi_D)\;. \quad (4)$$

- Second, the tangential speed of the gears ($r_D\frac{d\phi_D}{dt}$ and $r_F\frac{d\phi_F}{dt}$) at the contact point should equal, so we can remove dt and obtain

$$r_D(2\pi-\phi_D)\,d\phi_D \;=\; r_F(\pi-\phi_F)\,d\phi_F\;. \quad (5)$$

- In general, a full cycle (2π) of G_D should correspond to $1/K$ cycle of G_F in the periodic rotation, so we have

$$\psi(2\pi)=\frac{2\pi}{K}\;. \quad (6)$$

3.3. Computing Mating Gear Profiles

Next, we show that given a gear (with known geometry and rotation center) and also a transmission ratio K, the geometry and rotation center of its mating gear can be computed based on the above geometric and kinematic constraints; see Figure 6. Without loss of generality, we take the input gear as G_D.

(i) Locate the rotation center (L). By combining Equations 4 and 5, we can eliminate $r_F(\pi-\phi_F)$ and obtain

$$\frac{r_D(2\pi-\phi_D)}{L-r_D(2\pi-\phi_D)}=\frac{d\phi_F}{d\phi_D}=\psi'(\phi_D)\;. \quad (7)$$

Then, by replacing ϕ_D in Equation 7 with variable $\alpha\in[0,2\pi)$, and integrating both sides from 0 to ϕ_D, we have

$$\psi(\phi_D)=\int_0^{\phi_D}\frac{r_D(2\pi-\alpha)}{L-r_D(2\pi-\alpha)}d\alpha\;. \quad (8)$$

To satisfy the constraint in Equation 6, we set $\phi_D=2\pi$ in Equation 8 and then substitute the result into Equation 6 to obtain

$$\int_0^{2\pi}\frac{r_D(2\pi-\alpha)}{L-r_D(2\pi-\alpha)}d\alpha=\frac{2\pi}{K}\;. \quad (9)$$

Now, Equation 9 contains only one unknown, i.e., L. Observing that its left side always monotonically decreases with increase in L, we thus solve for L by performing a binary search in range $(r_D^{max},\,(K+1)r_D^{max}]$, where r_D^{max} is the maximum r_D for all θ_D. Then, we can locate the rotation center of the follower; see Figure 6(b).

(ii) Compute transmission function. With L, we further compute $\psi(\phi_D)$ by integrating Equation 8; see Figure 6(c) for an example.

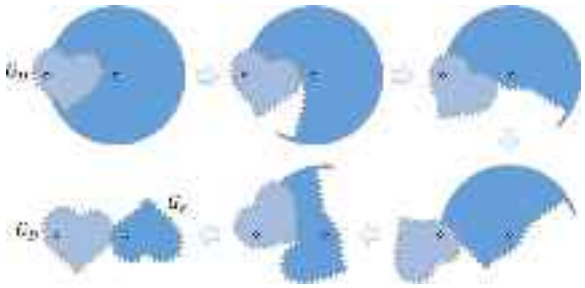


Figure 7: *The rotate-and-carve approach to generate G_F from G_D.*

Figure 8: *Mating gears (without teeth) computed using (a) rotate-and-carve and (b) analytical approaches, which take 7.33 and 0.34 seconds, respectively. Note that the gear in (b) is not functional due to part collision (see the red circle) during the rotation.*

(iii) Compute mating gear geometry. We initialize follower gear G_F as a circle of radius L, since it must be contained within such circle. Then, by rotating both G_D and G_F for a full cycle following $\psi(\phi_D)$ and using G_D's boundary to progressively carve the circle, we can obtain the shape of G_F; see Figure 7 for an example.

The above approach involves extensive constructive solid geometry computation. We thus resort to an analytical approach to compute G_F, for efficient evaluation when exploring gear designs. By Equation 4, the polar function of G_F, i.e., r_F, can be computed by

$$r_F(\pi-\phi_F)=L-r_D(2\pi-\psi^{-1}(\phi_F))\;, \quad (10)$$

where $\psi^{-1}(\phi_F)$ always exists because $\psi(\phi_D)$ is strictly monotonic. This approach computes G_F in $O(N)$, where N is the number of sample points on its boundary curve. It is worth to note that the resulting gear may not be functional, due to potential collisions with other parts in the driver. Figure 8 compares G_F produced by the two approaches. Note that we use the analytical approach for fast generation and evaluation of G_F when exploring the design space, and use the rotate-and-carve approach to produce G_F in the end.

4. Problem Formulation

We take two shapes, S_D and S_F, as well as transmission ratio K and target number of teeth M_D, as inputs. Our goal is to design a pair of functional gears G_D and G_F that resemble S_D and S_F, respectively. We formulate the design problem as an optimization that minimizes the following objective:

$$E=E_s\;+\;\omega\,E_\tau\;, \quad (11)$$

where E_s measures the amount of changes on S_D and S_F for forming the gear pair; E_τ is the maximal instantaneous torque required of G_D to drive the gears; and ω is a weight empirically set as 0.2. By using Equation 3, we can compute E_τ as $\max_{\phi_D}\tau_D(\phi_D)$. We will provide detail on how to compute E_s later.

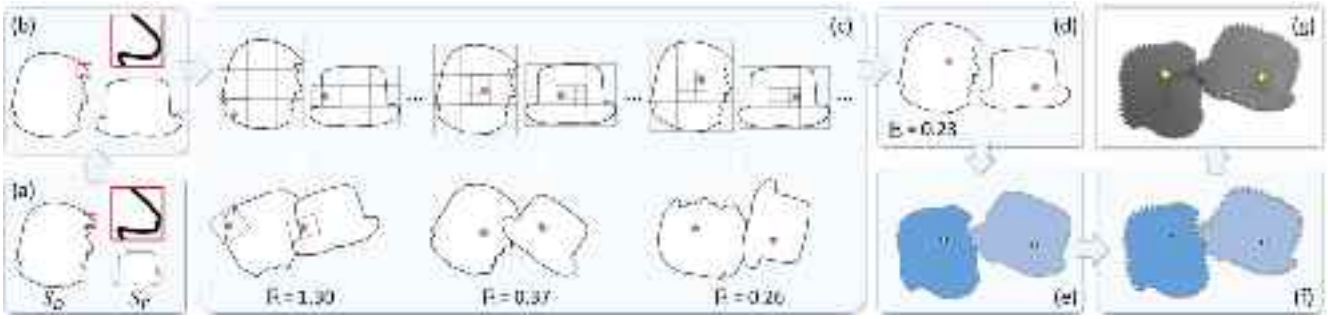


Figure 9: *Overview of our approach: (a&b) smooth and normalize the input shapes; (c) recursively sample and evaluate rotation center candidates; (d) choose the ones with minimal* E*; (e) construct the gear shapes; (f) generate teeth; and (g) fabricate the resulting gears.*

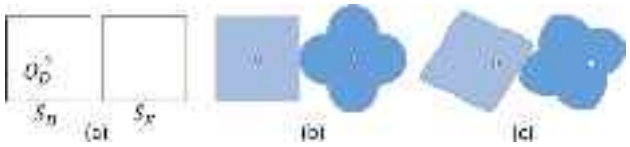


Figure 10: *(a) Input shapes with rotation center O_D marked on* S_D*. Mating gears generated using our (b) first and (c) second attempts.*

In general, the search space in the optimization consists of the boundary curves and rotation centers of both G_D and G_F, as well as the transmission function. However, since the parameters have to satisfy the constraints presented in Section 3, several of them are, in fact, dependent on one another. Hence, we need efficient strategies to explore the search space. Also, since the teeth geometry does not affect the objective, we perform teeth generation as a post-processing after solving the optimization; see Section 5.

Initial attempts. In our first attempt, we form the driver gear G_D by taking S_D as its boundary curve and selecting a point in S_D as its rotation center O_D, then compute G_D's mating gear G_F using the procedure in Section 3.3. This approach ensures G_D to have the same shape as S_D, but yet, the shape of the follower G_F may deviate a lot from S_F; see Figure 10(b) for a typical result.

Our second attempt addresses the above limitation by deforming both S_D and S_F, and measuring the shape change (i.e., E_s) by using a translational and rotational invariant shape descriptor [ARKF07]. However, there are no trivial methods (e.g., gradient-based) to guide the deformations measured by the descriptors, while ensuring that the generated gears form a mating pair. Hence, we resort to use a simulated annealing search [CDY98]; see Figure 10(c) for a typical result. However, the search easily gets stuck in local minima and is too time-consuming, due to the huge search space.

Our strategy. From Equations 7 and 4, we observe that the geometry of a mating gear pair (without teeth) can be fully determined by the transmission function and distance between rotation centers (i.e., L). Also, the rotation centers can locate at arbitrary points within the input shapes. These observations inspire us to develop an approach that first determines the rotation centers by using a sampling-based search and then computes the transmission function by taking the given shapes and selected rotation centers as inputs. Then, using the computed transmission function, we can reconstruct the gear shapes, which can be guaranteed to be a mating pair. Different from the two initial attempts, the shape change term E_s is indirectly minimized when computing the transmission function. Section 5 provides details about this approach.

5. Our Method

This section presents our method to solve the optimization problem formulated in Section 4. Since the input shapes S_D and S_F usually contain noise and sharp features, we first smooth them by using cubic spline interpolation [DBDBM*78]. Then, we normalize both shapes in a unit square and scale-up S_F, such that its perimeter is K times the perimeter of S_D; see Figure 9(a&b).

Figure 9 outlines our approach, which consists of three key steps. First, we develop a coarse-to-fine search to find a pair of optimal rotation centers that minimizes the objective in Equation 11 (see Figures 9(c&d)). Second, from the rotation centers and input shapes, we determine the transmission function that minimizes the shape change, and reconstruct the gear shapes from the transmission function accordingly (see Figure 9(e)). Third, we initialize teeth on gears and adjust the teeth height to avoid the gears from losing contacts during their rotations (see Figure 9(f)). Lastly, we fabricate the generated gears with teeth (see Figure 9(g)).

Note that to evaluate the quality of candidate rotation centers, the coarse-to-fine search in the first step relies on our mechanism formulated for the second step. Hence, we first present the second step in Section 5.1, followed by the coarse-to-fine search (first step) in Section 5.2 and teeth construction (third step) in Section 5.3.

5.1. Construct Mating Gears from Given Rotation Centers

In this section, we first consider the construction of mating gears with transmission ratio $K = 1$, then extend the method for $K > 1$.

Given shapes S_D and S_F with the corresponding rotation centers O_D and O_F, we first take S_D and O_D to form initial gear $\overline{G}_D$ and compute its transmission function following Equation 8, denoted as $\overline{\Psi}_D(\phi_D)$. Similarly, we can obtain initial follower gear $\overline{G}_F$ and transmission function $\overline{\Psi}_F(\phi_F)$; see Figure 11(b). If $\overline{G}_D$ and $\overline{G}_F$ are perfectly paired already, $\overline{\Psi}_D$ and $\overline{\Psi}_F$ should be an exact inverse of each other, i.e., $\overline{\Psi}_D = \overline{\Psi}_F^{-1}$, when we properly align them. Here, we denote ϕ_{shift} as the shift angle on $\overline{\Psi}_F$ to align $\overline{G}_F$ with $\overline{G}_D$.

In general, it is likely that the initial gears $\overline{G}_D$ and $\overline{G}_F$ do not perfectly pair; see Figure 11(c). Hence, to deform them and make them a mating pair, we design the following procedure:

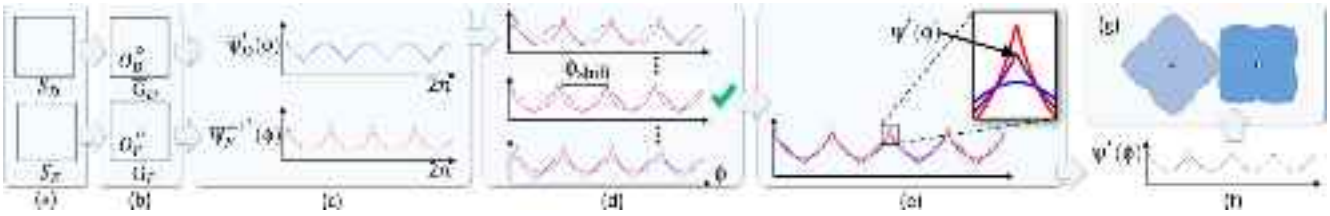


Figure 11: *Our approach to compute mating gears from (a) input shapes. We first (b) locate a rotation center in each input shape, (c) compute the derivatives of their transmission functions, (d) find the shift angle to align the two derivatives, (e&f) obtain the derivative of the target transmission function by interpolating the two aligned derivatives, and lastly, (g) generate boundary curves (shapes) of the resulting gears.*

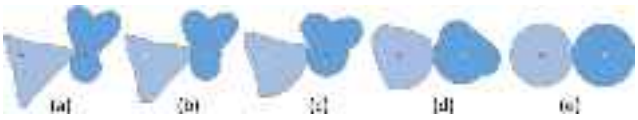


Figure 12: *Gears generated (a) without and (b)-(e) with the optional step of bounding the maximum torque, in which the maximum instantaneous torque of the driver does not exceed (b)* 3.0 τ_F*, (c)* 2.0 τ_F*, (d)* 1.2 τ_F *and (e)* 1.0 τ_F*. Here, we set* $K = 1$.

(i) Align initial gears. To align the initial gears, we need to find a proper orientation (i.e., ϕ_{shift}) to pose $\overline{\text{G}}_F$ relative to $\overline{\text{G}}_D$. To obtain ϕ_{shift}, we shift $\overline{\text{G}}_F$'s transmission function ($\overline{\psi}_F$) for different angles, and try to match it with $\overline{\text{G}}_D$'s transmission function ($\overline{\psi}_D$). Since derivative is independent of the starting point (i.e., the gears' initial orientation) and is more sensitive to changes, we thus compute the transmission function derivatives $\overline{\psi}'_D$ and $\overline{\psi}_F^{-1'}$, and use them to find the shift angle that minimizes their difference:

$$\Delta = \int_0^{2\pi} (\overline{\psi}'_D(\phi) - \overline{\psi}_F^{-1'}(\phi + \phi_{\text{shift}}))^2 d\phi \,. \tag{12}$$

Here, we uniformly sample M candidate values of ϕ_{shift} in $[0, 2\pi]$, and empirically set $M = 128$ in our experiments. Then, we use Equation 12 to evaluate Δ for each candidate, and choose the one with the least Δ (denoted as Δ_{min}) as ϕ_{shift}; see Figure 11(d).

Interestingly, Δ_{min} actually indicates the amount of deformation (i.e., the change of r(θ) along the radial direction θ) needed on the input shapes to make them a mating gear pair. Hence, we take it to model the E_s term in our objective (see Equation 11):

$$\text{E}_s = \Delta_{\text{min}} \,. \tag{13}$$

The underlying reason is that when the rotation centers are given, L is known, and therefore, the gear shape (r_D) and transmission function derivative (ψ') are actually one-to-one corresponding; see Equation 7. This is also evidenced by the fact that when $\Delta_{\text{min}} = 0$, the two transmission function derivatives ($\overline{\psi}'_D$ and $\overline{\psi}_F^{-1'}$) perfectly aligns with each other, and the two shapes are already a mating pair without the need to deform them. By this one-to-one correspondence, we thus minimize Δ_{min} to minimize shape changes (E_s).

(ii) Compute target transmission function. Next, we shift $\overline{\psi}_F^{-1'}$ by ϕ_{shift}, and compute the derivative of the target transmission function $\psi'(\phi)$ (see Figure 11(e&f)) by interpolating $\overline{\psi}'_D$ and $\overline{\psi}_F^{-1'}$:

$$\psi'(\phi) = t\overline{\psi}'_D(\phi) + (1-t)\overline{\psi}_F^{-1'}(\phi + \phi_{\text{shift}}) \,, \tag{14}$$

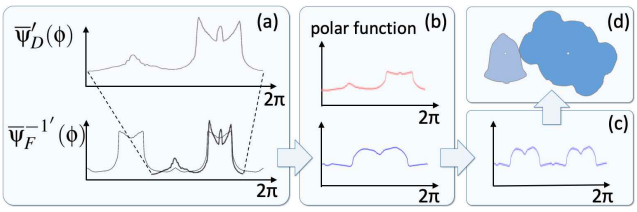


Figure 13: *Constructing mating gears for* $K = 2$*. We match (a)* $\overline{\psi}'_D(\phi)$ *with* $1/K$ *of* $\overline{\psi}_F^{-1'}(\phi)$*, (b&c) duplicate the resulting polar function (in blue) of the follower gear K times, normalize it, and then (d) compute the boundary curves of the gears.*

where $t \in [0, 1]$ is a user-specified parameter and its default value is set as 0.5 to balance the deformations on the two input shapes. Note that $\psi'(\phi)$ will later be used to reconstruct the shape of both the driver and follower gears, so by the above interpolation, we can make $\psi'(\phi)$ to be similar to both $\overline{\psi}'_D$ and $\overline{\psi}_F^{-1'}$. Therefore, the resulting gear shapes can better resemble the inputs with minimized shape changes. Furthermore, we can use t to control how close the resulting gears resemble each input shape.

(iii) Compute gear boundary curves. Using $\psi'(\phi)$, O_D, and O_F, we can compute the boundary curves of G_D and G_F using Equation 7, and then Equation 4; see Figure 11(g) for a typical result. Comparing the result with those by the initial attempts (see Figure 10(b&c)), this constructive approach produces gears with smaller and more balanced modifications.

Optional step: bound the maximum torque. By combining Equations 3, 4, and 7, we can derive

$$\psi'(\phi_D) = \tau_D(\phi_D)/\tau_F \,, \tag{15}$$

meaning that the instantaneous torque $\tau_D(\phi_D)$ to drive the gears can be measured by $\psi'(\phi_D)$ as τ_F is a constant. Considering that motors usually have a limit on the maximum output torque, we allow users to set a threshold τ_{thres} to bound the driver's torque $\tau_D(\phi_D)$. Hence, if τ_{thres} has been set, we truncate $\psi'(\phi_D)$ from Equation 14 by

$$\psi'_{\text{trunc}}(\phi_D) = \begin{cases} \tau_{\text{thres}}/\tau_F & \text{if } \psi'(\phi_D) > \tau_{\text{thres}}/\tau_F \\ \psi'(\phi_D) & \text{otherwise} \,. \end{cases} \tag{16}$$

Further, we scale $\psi'_{\text{trunc}}(\phi_D)$ to satisfy $\int_0^{2\pi} \psi'_{\text{trunc}}(\phi_D) d\phi_D = 2\pi$, since transmission functions have to produce a full cycle on the follower after the driver makes a full turn, for the case of $K = 1$. Figure 12 shows gears generated using different τ_{thres}, in which the gear shapes become more circular when τ_{thres} decreases.

Extension: transmission ratio $K \neq 1$. We need three adaptations to our approach to construct mating gears when $K \neq 1$. In step (i), we match $\overline{\psi}'_D(\phi)$ in range $[0, 2\pi]$ with $1/K$ of $\overline{\psi}_F^{-1'}(\phi)$ in range $[0, 2\pi/K]$ to find ϕ_{shift} using Equation 12; see Figure 13(a). In step (ii), we normalize the selected part of $\overline{\psi}_F^{-1'}(\phi)$ from $[0, 2\pi/K]$ to $[0, 2\pi]$. After that, we use the original $\overline{\psi}'_D(\phi)$ and normalize $\overline{\psi}_F^{-1'}(\phi)$ to compute $\psi'(\phi)$ according to Equation 14. In step (iii), after obtaining the polar function of the follower based on $\psi'(\phi)$, we further duplicate the polar function K times and normalize it to create the final polar function; see Figure 13 (b&c).

5.2. Coarse-to-fine Search for Optimal Rotation Centers

Given a pair of rotation centers, the above constructive approach not just generates a pair of mating gears, but also provides a quantitative measure E_s (or $\Delta_{\min}$) on the amount of required changes to the input shapes. Hence, the domains of the rotation center pair (O_D and O_F) can be regarded as the search space, since other parameters can be computed from them. Therefore, we apply the constructive approach to adaptively explore the search space by sampling candidate rotation center pairs and evaluating their quality using the objective E to locate the rotation center pair that minimizes E.

To speed up the search, we sample rotation center pairs inside S_D and S_F in a coarse-to-fine manner. In detail, for each input shape, we first construct a 3×3 grid within its bounding box, locate the centers in the grid cells, and discard centers that are outside the shape boundary; see the top-left illustration in Figure 9(c). Here, we denote $\{O_D^i\}$ as the list of remaining rotation center candidates in shape S_D, and likewise, $\{O_F^j\}$ for S_F. Next, we try all possible pairs of O_D^i and O_F^j as candidate rotation center pairs to construct mating gears (using the constructive approach in Section 5.1) but stop the computation when we obtain the E_s and objective function value E. In this way, we can locate the best J rotation center pairs with the smallest E. After that, we subdivide each cell associated with the best rotation centers into a 3x3 grid, gather the new set of cell centers within the shape as $\{O_D^i\}$ and $\{O_F^j\}$, then repeat the above evaluation process. By recursively performing the above process three times, we can obtain the rotation center pair that minimizes E and take them as O_D and O_F; see Figure 9(c) & (d) for a running example. Note that we empirically set $J = 3$ in our implementation.

Once the optimal rotation centers O_D and O_F are found, we apply our constructive approach on them to compute the shapes of the mating gears; see Figure 9(e).

5.3. Constructing Teeth on Mating Gears

To add teeth on the constructed gear shapes, we first locate M_D uniformly-spaced sample points on the driver shape's boundary, and use the standard involute tooth profile [LFA04] to initialize $\{h^i\}$ and $\{s^i\}$, and construct teeth on the driver gear. Then, taking ρ as the tolerance between gears, we apply the rotate-and-carve technique (see Figure 7) to generate teeth on the follower shape; see Figure 14(a&b). However, we found that the resulting gears can easily lose contacts after some rotation; see Figure 14(c). This is because some of the teeth's local moving direction is simply not blocked by the corresponding teeth on the follower gear.

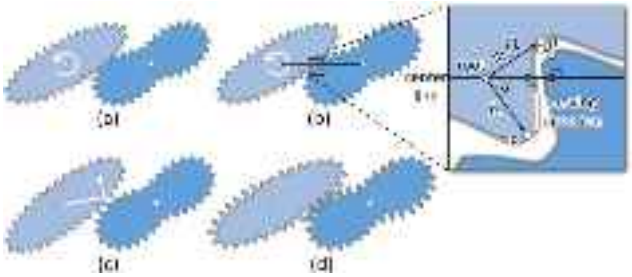


Figure 14: *Constructing teeth on gears (a) without and (b) with tolerance, showing that the tolerance is small. (c) Gears with regular teeth could lose contact during motion transfer. (d) We create customized teeth with non-uniform shape for better gear meshing.*

To resolve the problem, we individually scale the height (h^i) of each tooth, such that when it rotates, it can always push the corresponding tooth on the follower and make it move. Mathematically, for each tooth in the driver gear, we consider the moment at which the root of the tooth just passes through the center line between O_D and O_F (see the inset in Figure 14(b)). Then, we formulate the following inequality that describes the situation that the tooth can contact and push the corresponding tooth:

$$h_i \cos\alpha \geq \frac{s_i}{2} \sin\alpha + \rho \cos\alpha \,. \tag{17}$$

Hence, we set h_i as $\frac{s_i}{2}\tan\alpha + \rho$ for constructing the tooth on the driver. In the end, we generate a 3D mesh models for each gear by triangulating the 2D gear profile and thickening it, and then, fabricate the gear models using 3D printing; see Figure 9(g).

6. Results and Experiments

We implement our method in Python 3.7 and run it on a laptop with a dual-core Intel i5 CPU and 16GB RAM. We use NumPy [Oli19] to manipulate the arrays and their computation, and use Shapely [G*19] for the basic geometric computing, such as CSG operations and triangulation. To generate the input 2D shapes, we compile a collection of shapes, including 28 silhouette images and 84 images selected from two public silhouette image datasets, i.e., MPEG-7 database [LLE00] and the Animal database from [BLT09]. Then, we manually select 300 pairs of shapes from the collection as inputs to our method. Here, we consider three selection criteria: (i) the shapes in each gear pair should have relevant semantic meaning; (ii) shapes with delicate structures, e.g., long and thin tails, are not selected; and (iii) we avoid overly using the same shape. For each pair of input shapes, we try our method with three different values of transmission ratio K, i.e., K = 1, 2, and 3. We sort all the results in ascending order of the resulting objective function value E, and report the top ones in the paper.

Generated gears of different variety. Figure 15 shows 14 pairs of non-circular gears of various shapes generated by our method; see also the supplementary video for animated results. When the two input shapes are the same, the generated gear pair will have the same boundary curve, if we set $t = 0.5$ in Equation 14 for generating the resulting gear shapes; see the top two rows in Figure 15. Also, our method allows users to express preference on which of the two input shapes to be better preserved in the resulting gears

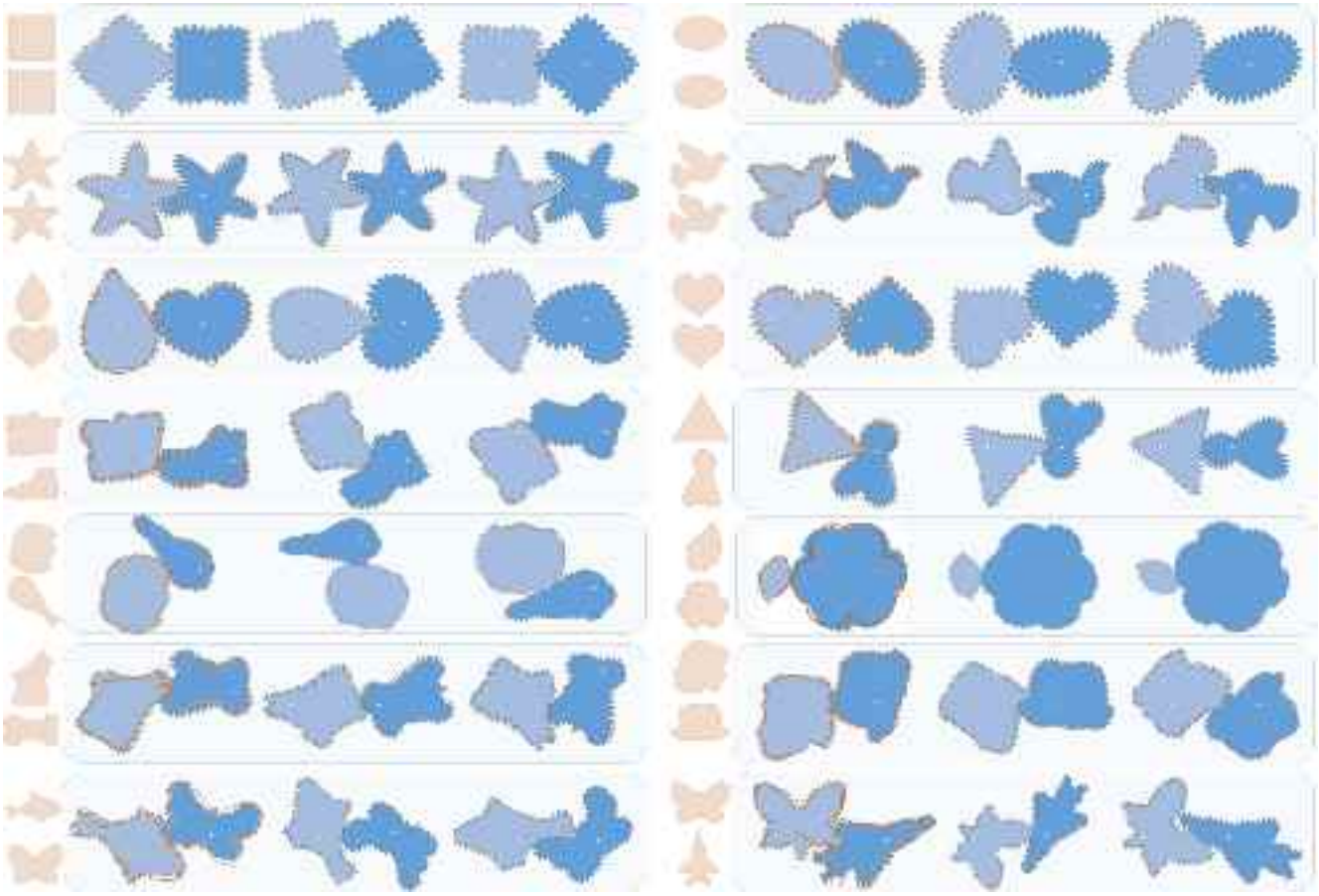


Figure 15: *A gallery of non-circular gears computed by our method. From top to bottom: left column:* SQUARE-SQUARE, STARFISH-STARFISH, DROP-HEART, POT-SHOE, FISH-BUTTERFLY, DOG-BONE, TRUMP-CLUB*; right column:* ELLIPSE-ELLIPSE, TRIANGLE_BELL-SUNNY_DOLL, LEAF-FLOWER, DOVE-DOVE, HAT-GIRL, BUTTERFLY-FIGHTER. *For each generated gear pair, we show the input shapes on the left, and three snapshots of rotating gears on the right. We overlay the input shape to gears in the first snapshot to show the deformation.*

Figure 16: *(a) Given the* BELL-CANDY *input, three sets of results are generated for different transmission ratios: (b)* $K = 1$*, (c)* $K = 2$ *and (d)* $K = 3$*. The corresponding objective function value is (b)* $\mathrm{E} = 0.30$*, (c)* $\mathrm{E} = 0.24$*, and (d)* $\mathrm{E} = 0.43$*.*

by simply adjusting the value of t; see two typical examples in Figure 17. Note further that preserving more of one input shape will deteriorate the shape of the other, and vice-versa.

In our experiments, we found that many desirable results are generated for $K = 1$ but not for $K > 1$; see again Figure 15. This is because $K > 1$ requires an input shape of K-fold rotational symmetry. This is not common in practice; see the LEAF-FLOWER result with $K = 3$ in Figure 15 (5th row, right column) for an example. Figure 16 shows three pairs of BELL-CANDY gears generated for the same input shapes but using different transmission ratios (i.e., $K = 1, 2, 3$) as well as the corresponding objective function value E, and we can see that significant shape change happens for $K = 1, 3$.

Statistics. Table 1 shows the statistics of our results. Overall, our method can generate a gear pair in around 9.7 minutes on average, which can be significantly sped up with a C/C++ implementation in the future. As expected, the rotation center search and the rotate-and-carve procedure take up most of the processing time, which are 9.5 and 0.15 minutes on average, respectively. Also, we observe that input shapes of larger area (e.g., SQUARE-SQUARE) usually need more time to process, since there are more rotation center candidates to be evaluated inside the input shapes.

Fabrication. We use an FDM 3D printer (Ultimaker 2 Extended plus) with printing resolution 0.1mm and an SLA printer (Form 2) with printing resolution 0.05mm to print our designed gears. To accommodate the manufacturing inaccuracy, we shrink the boundary of each gear (driver and follower) by $0.1mm$, so the overall tolerance is $0.2mm$. We assemble a LEGO Technic test bed to examine the usability of the generated gears. To do so, we first scale the gear pairs, such that the distance between their rotation centers equals to $N \times 7.76mm$, where N is a positive integer and $7.76mm$ is the size

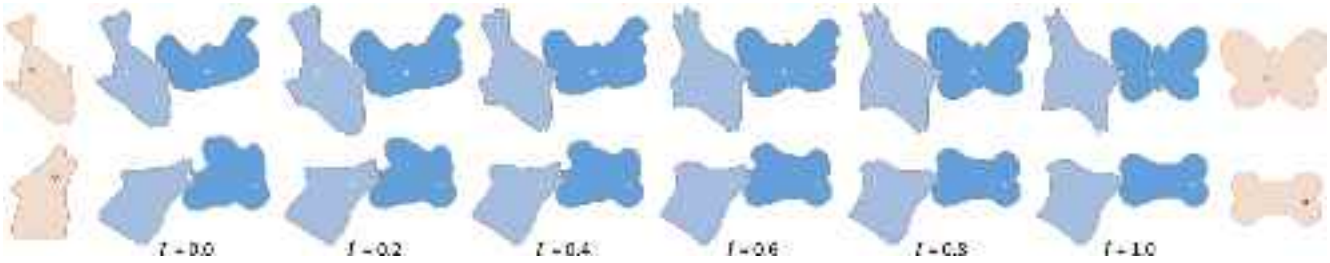


Figure 17: *Preference on preserving one particular input shape can be expressed by adjusting the value of t in Equation 14. Note that the rotation centers of the gears are fixed in these optimizations (see the dots in the input shapes). Top:* FISH-BUTTERFLY*; bottom:* DOG-BONE.

Table 1: *Statistics of our results. From left to right, the name of input shape, transmission ratio K, number of teeth M_D on the driver gear, number of vertices $V(G_F)$ of the follower gear model (note that the number of vertices of the driver gear is* 1024 *for all the results), pre-processing time, time to find the optimal rotation centers, time to generate the follower gear using the rotate-and-carve technique, and total time to generate the gears.*

Model	Input & Output			Running time (minutes)			
	K	M_D	$V(G_F)$	Pre-process	Rotation center search	Rotate-and-carve	Total
SQUARE-SQUARE	1	32	1535	0.015	12.83	0.08	12.92
ELLIPSE-ELLIPSE	1	32	1107	0.005	7.65	0.06	7.71
STARFISH-STARFISH	1	64	3388	0.006	8.77	0.16	8.93
DOVE-DOVE	1	64	4275	0.013	8.55	0.21	8.77
HEART-HEART	1	32	2933	0.013	10.12	0.12	10.25
TRIANGLE_BELL-SUNNYDOLL	1	48	4475	0.010	8.65	0.21	8.87
DROP-HEART	1	32	1122	0.013	10.30	0.07	10.38
BELL-CANDY	2	32	5033	0.008	8.31	0.24	8.55
LEAF-FLOWER	3	32	8608	0.008	10.37	0.14	10.51
POT-SHOE	1	64	3776	0.010	8.28	0.20	8.49
FISH-BUTTERFLY	1	48	2462	0.018	12.07	0.12	12.21
DOG-BONE	1	64	4912	0.008	8.17	0.22	8.39
GIRL-HAT	1	64	2884	0.010	9.22	0.14	9.37
TRUMP-CLUB	1	48	1898	0.010	5.81	0.11	5.93
BUTTERFLY-FIGHTER	1	64	3904	0.015	9.58	0.21	9.80
HAT-TRUMP	1	64	2907	0.013	10.36	0.13	10.50
BOY-GIRL	1	64	3570	0.017	12.50	0.18	12.70

of a grid cell in our test bed. Second, we create a cross-shaped hole at the rotation center of the 3D-printed gears, so that we can assemble it with the LEGO® Technic axle beam. Using this testbed, we can put together our 3D-printed gears, and evaluate the gears usability by connecting the axle of the driver gear to the LEGO® motor. Figure 18 (top) shows six pairs of gears that are in motion.

Besides, we also build a train of twelve gears, which consists of six pairs of gears with exactly the same shapes that are connecting alternatively; see Figure 18 (bottom). Please watch the supplementary video for the animated results.

Experiment on torque. Our method not only minimizes the torque (i.e., E_τ) required to drive the gears, but also allows users to bound the driver's torque by setting the threshold τ_{thres}. We conducted two experiments to evaluate these two features, respectively. In the first experiment, we generate two pairs of HEART-HEART gears with different ω in our objective function (see Equation 11), i.e., $\omega = 0.2$ and $\omega = 1.0$, respectively. Figure 19 (b&c) shows the

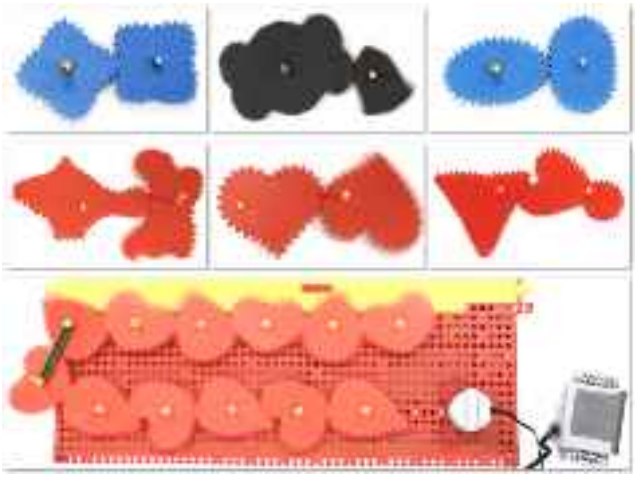


Figure 18: *Our 3D-printed non-circular gears. Top:* SQUARE-SQUARE*,* BELL-CANDY *&* ELLIPSE-ELLIPSE*; middle:* FISH-BUTTERFLY*,* HEART-HEART *&* TRIANGLE_BELL-SUNNY_DOLL*; and bottom: a gear train composed of six pairs of* DROP-HEART.

two generated gear pairs. We can see from the figure that the rotation centers of the gear pair with $\omega = 1.0$ (Figure 19 (c)) locates more close to the shape center, but the gear boundary is more deformed from the inputs. This is because a larger ω emphasizes more on minimizing the maximum torque in the gear rotation. This is equivalent to putting the rotation center at the shape centroid. In the second experiment, we generate two pairs of BOY-GIRL gears with and without having $\tau_{\text{thres}} = 2\tau_F$; see Figure 19 (e&f).

To experimentally compare the generated gears, we create a simple system to measure the maximum torque required to drive the 3D-printed gears. First, we use a LEGO Technic pin with the friction ridges (constant friction torque of 0.01195 Nm) to block the follower gear to rotate, which is employed to simulate a load acted on the follower gear; see the blue axle shown in Figure 19. Next, we use a force gauge to measure the force with a 23.9mm arm of force, which has a maximum range of 50N and an accuracy of 0.25N, and is able to record the maximum force it measured. After the preparation work, we divide a full rotation cycle of the driver into 16 samples, measure the required driving force for each sample, and record the maximum force. Figure 19 shows the maximum driving force and the corresponding gear poses, thereby verifying the effectiveness of our method to reduce the required driving torque. Please watch the supplementary video for more detail.

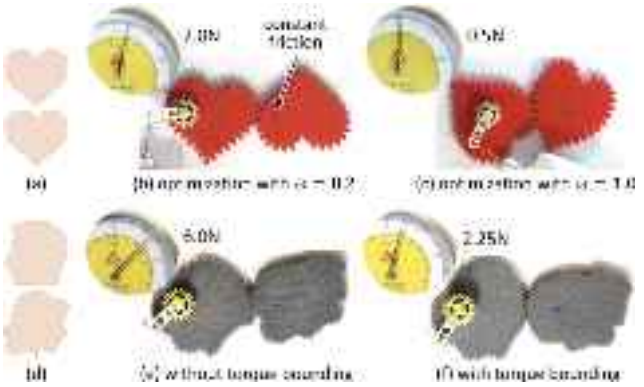


Figure 19: *Experiments on maximum instantaneous torque required to drive the gears. From the same input shapes (a&d),* HEART-HEART *gears generated with (b)* $\omega = 0.2$ *and (c)* $\omega = 1.0$*; and* BOY-GIRL *results generated (e) without and (f) with torque bounding. The torque is measured by a force gauge with a fixed arm of force (see the numbers in the yellow dial plate).*

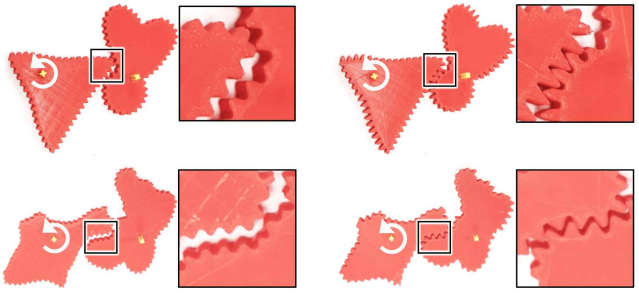


Figure 20: *Gears generated with regular and customized teeth. Left: gears with regular teeth could lose contact during motion transfer (see the zooming views). Right: gears with customized teeth could maintain unceasing contact.*

Figure 21: *Two typical failure cases. Left: not all 2D shapes can be represented by a polar function (range* $[0, 2\pi)$*); see the red and green dots. Right: our method cannot preserve very fine geometric features on the input shapes, as such details cannot be well encoded in the transmission functions for generating the gears.*

Experiment on teeth. To test the effectiveness of our customized teeth, we generate two pairs of gears with the same boundary curve but different teeth, i.e., regular and customized teeth, where the customized teeth are generated using the approach presented in Section 5.3. For each gear pair, we rotate the driver gear and observe the reaction of the follower gear. In our experiments, the gears with regular teeth could lose contact at a certain moment; see Figure 20 (left). In contrast, the gears with customized teeth can transfer motion continuously without such issue; see Figure 20 (right). Please watch the supplementary video for animated results.

7. Conclusion

We present a new method to compute and generate non-circular gears that are functional and resemble the given 2D shapes. The foundation of our method is a discrete geometric characterization of the non-circular gear, as well as the kinematic and dynamic modeling built on top of it. Inspired by the close relationship between gear geometry and kinematics, we formulate a constructive approach that takes the transmission functions of the input shapes as an intermediary, and then minimally modifies the two input shapes to form a mating gear pair with fixed rotation centers. One key innovation is to work with the derivative of the transmission functions to solve for a pair of optimal rotation centers, then to take these rotation centers to directly reconstruct the gear shapes. In this way, we can avoid explicit and tedious 2D shape deformations, yet being able to modify the input shapes with minimized shape changes. Our method can produce non-circular gears of a rich variety of shapes that were not possible before. Also, these gears require only a relatively small torque to drive, as verified in our physical experiments.

Potential applications. The non-circular gears generated from our framework seek visually-appealing results on top of a functionality guarantee, enabling us to apply them in various usage scenarios. For example, we can employ them as components in decorative artworks and mechanical devices, combining form and function in an aesthetically-pleasing manner. Besides, we can use non-circular gears of shape optimized for assorted functionality requirements, e.g., avoiding potential collision with the enclosure and with the other parts in a mechanical or robotic system.

Discussion. Our work has some limitations that could inspire interesting directions for future research. First, our method may not work well for arbitrary inputs, meaning that the generated gears cannot be faithful to the inputs. Figure 21 shows two typical cases, where the input shapes cannot be represented by polar functions (left), and they contain characteristic features that are lost during the gear generation (right). Second, our current implementation considers only shape similarity and maximum torque. It would be helpful to include other measures, such as aesthetics (e.g., maintain shape features), structural strength, kinematic and dynamic performance, etc. Third, the decomposition of gear geometry into boundary curve and teeth profile could result in non-fabricable geometry; e.g., teeth added on thin gear parts could create a gear with disconnected geometry. Last, we focus on designing 2D non-circular gear pairs. Further work to explore the construction of non-circular gears with other kinds of mechanical elements (e.g., linkages), as well as non-circular gears in 3D space, to form mechanisms for performing more complex tasks would be interesting.

Acknowledgements. This work is supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK 14201918 and 14203416), SUTD Start-up Research Grant (Award Number: SRG ISTD 2019 148), ERC Starting Grant, ERC PoC Grant, Google Faculty Award, Royal Society Advanced Newton Fellowship, and gifts from Adobe.

References

[AB10] ARAKELIAN V., BRIOT S.: Simultaneous inertia force/moment balancing and torque compensation of slider-crank mechanisms. *Mechanics Research Communications 37*, 2 (2010), 265–269. 2

[ARKF07] ALAJLAN N., RUBE I. E., KAMEL M. S., FREEMAN G.: Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition 40*, 7 (2007), 1911–1920. 5

[Bai02] BAIR B.-W.: Computer aided design of elliptical gears. *Journal of Mechanical Design 124*, 4 (2002), 787–793. 1

[Bäs19] BÄSEL U.: Determining the geometry of noncircular gears for given transmission function. *arXiv preprint arXiv:1905.02642* (2019). 2

[BCT15] BÄCHER M., COROS S., THOMASZEWSKI B.: LinkEdit: Interactive linkage editing using symbolic kinematics. *ACM Trans. on Graph. (SIGGRAPH) 34*, 4 (2015). Article 99. 2

[BLT09] BAI X., LIU W., TU Z.: Integrating contour and skeleton for shape classification. In *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops* (2009), IEEE, pp. 360–367. 7

[CDY98] CAGAN J., DEGENTESH D., YIN S.: A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout. *Computer-aided design 30*, 10 (1998), 781–790. 5

[CLM*13] CEYLAN D., LI W., MITRA N. J., AGRAWALA M., PAULY M.: Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. on Graph. (SIGGRAPH Asia) 32*, 6 (2013). Article No. 186. 2

[CTN*13] COROS S., THOMASZEWSKI B., NORIS G., SUEDA S., FORBERG M., SUMNER R. W., MATUSIK W., BICKEL B.: Computational design of mechanical characters. *ACM Trans. on Graph. (SIGGRAPH) 32*, 4 (2013). Article no. 83. 1, 2

[DBDBM*78] DE BOOR C., DE BOOR C., MATHÉMATICIEN E.-U., DE BOOR C., DE BOOR C.: *A practical guide to splines*, vol. 27. springer-verlag New York, 1978. 5

[DYYT17] DUNCAN N., YU L.-F., YEUNG S.-K., TERZOPOULOS D.: Approximate dissections. *ACM Trans. on Graph. (SIGGRAPH Asia) 36*, 6 (2017). Article No. 182. 2

[FLC00] FIGLIOLINI G., LANNI C., CECCARELLI M.: On the kinematic synthesis of non-circular gears. In *International Conference on Gears, Transmissions and Mechanical Systems* (2000), pp. 283–292. 2

[G*19] GILLIES S., ET AL.: Shapely: Manipulation and analysis of geometric objects, 2019. https://github.com/Toblerity/Shapely. 7

[HL15] HERGEL J., LEFEBVRE S.: 3d fabrication of 2d mechanisms. *Computer Graphics Forum 34*, 2 (2015), 229–238. 2

[KB99] KAHRAMAN A., BLANKENSHIP G. W.: Effect of involute contact ratio on spur gear dynamics. *Journal of Mechanical Design 121*, 1 (1999), 112–118. 1

[KNF*17] KARPOV O., NOSKO P., FIL P., NOSKO O., OLOFSSON U.: Prevention of resonance oscillations in gear mechanisms using non-circular gears. *Mechanism and Machine Theory 114* (2017), 1–10. 2

[KSH*16] KWAN K. C., SINN L. T., HAN C., WONG T.-T., FU C.-W.: Pyramid of arclength descriptor for generating collage of shapes. *ACM Trans. on Graph. (SIGGRAPH Asia) 35*, 6 (2016). Article No. 229. 2

[Lac08] LACZIK B.: Design of profile of the non-circular gears, 2008. G-2008-A-08. 2

[Law13] LAW B.: Wooden clocks with non-circular gears, 2013. https://www.youtube.com/watch?v=IzznSFmz5rM. 2

[LFA04] LITVIN F. L., FUENTES-AZNAR A.: *Gear Geometry and Applied Theory*. Cambridge University Press, 2004. 1, 3, 7

[LFAGPH09] LITVIN F. L., FUENTES-AZNAR A., GONZALEZ-PEREZ I., HAYASAKA K.: *Noncircular Gears: Design and Generation*. Cambridge University Press, 2009. 1, 2, 3

[LLE00] LATECKI L. J., LAKAMPER R., ECKHARDT T.: Shape descriptors for non-rigid shapes with a single closed contour. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)* (2000), vol. 1, IEEE, pp. 424–429. 7

[LMAH*18] LI S., MAHDAVI-AMIRI A., HU R., LIU H., ZOU C., VAN KAICK O., LIU X., HUANG H., ZHANG H.: Construction and fabrication of reversible shape transforms. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018). Article No. 190. 2

[MLB*09] MODLER K.-H., LOVASZ E.-C., BÄR G., NEUMANN R., PERJU D., PERNER M., MĂRGINEANU D.: General method for the synthesis of geared linkages with non-circular gears. *Mechanism and Machine Theory 44*, 4 (2009), 726–738. 2

[Mun06] MUNDO D.: Geometric design of a planetary gear train with non-circular gears. *Mechanism and Machine Theory 41*, 4 (2006), 456–472. doi:https://doi.org/10.1016/j.mechmachtheory.2005.06.003. 2

[MZB*17] MEGARO V., ZEHNDER J., BÄCHER M., COROS S., GROSS M., THOMASZEWSKI B.: A computational design tool for compliant mechanisms. *ACM Trans. on Graph. (SIGGRAPH) 36*, 4 (2017). Article No. 82. 2

[NBA19] NISHIDA G., BOUSSEAU A., ALIAGA D. G.: Multi-pose interactive linkage design. *Computer Graphics Forum (Eurographics) 38*, 2 (2019), 277–289. 2

[Oli19] OLIPHANT T.: NumPy: The fundamental package for scientific computing with Python, 2019. http://www.numpy.org/. 7

[RCLM18] ROUSSEL R., CANI M.-P., LÉON J.-C., MITRA N. J.: Exploratory Design of Mechanical Devices with Motion Constraints. *Computers and Graphics 74* (2018), 244–256. 2

[SKAM17] SAPUTRA R. A., KAPLAN C. S., ASENTE P., MĚCH R.: FLOWPAK: Flow-based ornamental element packing. In *Graphics Interface Conference* (2017), pp. 8–15. 2

[SWT*17] SONG P., WANG X., TANG X., FU C.-W., XU H., LIU L., MITRA N. J.: Computational design of wind-up toys. *ACM Trans. on Graph. (SIGGRAPH Asia) 36*, 6 (2017). Article No. 238. 2

[TCG*14] THOMASZEWSKI B., COROS S., GAUGE D., MEGARO V., GRINSPUN E., GROSS M.: Computational design of linkage-based characters. *ACM Trans. on Graph. (SIGGRAPH) 33*, 4 (2014). Article No. 64. 2

[Val19] VALLE M.: Gearify software, 2019. http://www.gearifysoftware.com. 2

[Whe07] WHEEL M.: Elliptical gears, 2007. https://www.youtube.com/watch?v=RqDTwPla6bo. 2

[XKCB18] XU H., KNOOP E., COROS S., BÄCHER M.: Bend-It: Design and fabrication of kinetic wire characters. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018). Article No. 239. 2

[ZAC*17] ZHANG R., AUZINGER T., CEYLAN D., LI W., BICKEL B.: Functionality-aware retargeting of mechanisms to 3D shapes. *ACM Trans. on Graph. (SIGGRAPH) 36*, 4 (2017). Article No. 81. 2

[ZCR*16] ZOU C., CAO J., RANAWEERA W., ALHASHIM I., TAN P., SHEFFER A., ZHANG H.: Legible compact calligrams. *ACM Trans. on Graph. (SIGGRAPH) 35*, 4 (2016). Article No. 122. 2

[ZHH*16] ZHENG F., HUA L., HAN X., LI B., CHEN D.: Synthesis of indexing mechanisms with non-circular gears. *Mechanism and Machine Theory 105* (2016), 108–128. 1, 2

[ZW12] ZHOU Y., WANG R.: An algorithm for creating geometric dissection puzzles. In *Bridges Towson: Mathematics, Music, Art, Architecture, Culture* (2012), pp. 49–56. 2

[ZXS*12] ZHU L., XU W., SNYDER J., LIU Y., WANG G., GUO B.: Motion-guided mechanical toy modeling. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012). Article No. 127. 2