

Exact 3D Path Generation via 3D Cam-Linkage Mechanisms

YINGJIE CHENG, University of Science and Technology of China, China and Singapore University of Technology and Design, Singapore

PENG SONG, Singapore University of Technology and Design, Singapore

YUKUN LU, University of Science and Technology of China, China and Singapore University of Technology and Design, Singapore

WEN JIE JEREMY CHEW, Singapore University of Technology and Design, Singapore

LIGANG LIU, University of Science and Technology of China, China

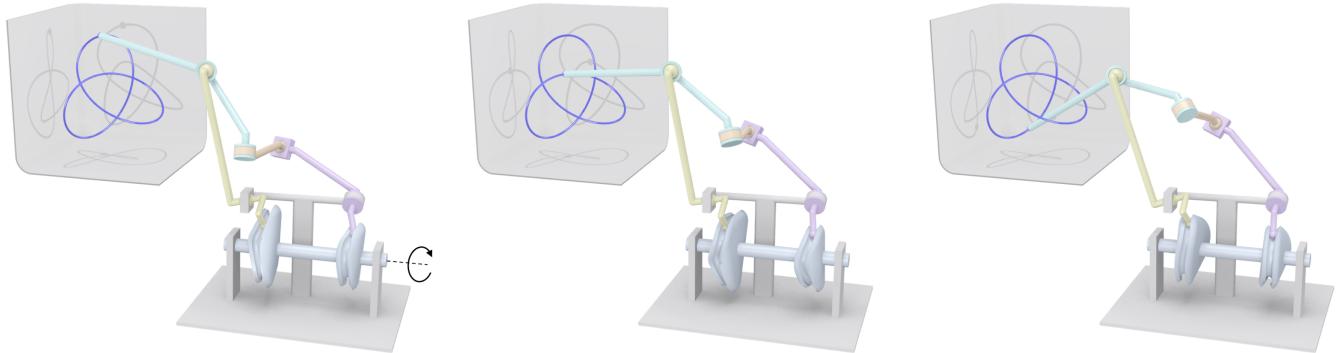


Fig. 1. We present a new 3D cam-linkage mechanism that can exactly generate a user-specified 3D path such as the TREFOIL KNOT curve shown here, driven by a single actuator (indicated by a circular arrow on the left).

Exact 3D path generation is a fundamental problem of designing a mechanism to make a point *exactly* move along a prescribed 3D path, driven by a single actuator. Existing mechanisms are insufficient to address this problem. Planar linkages and their combinations with gears and/or plate cams can only generate 2D paths while 1-DOF spatial linkages can only generate 3D paths with rather simple shapes. In this paper, we present a new 3D cam-linkage mechanism, consisting of two 3D cams and five links, for exactly generating a continuous 3D path. To design a 3D cam-linkage mechanism, we first model a 3-DOF 5-bar spatial linkage to exactly generate a prescribed 3D path and then reduce the spatial linkage's DOFs from 3 to 1 by composing the linkage with two 3D cam-follower mechanisms. Our computational approach optimizes the 3D cam-linkage mechanism's topology and geometry to minimize the mechanism's total weight while

ensuring smooth, collision-free, and singularity-free motion. We show that our 3D cam-linkage mechanism is able to exactly generate a continuous 3D path with arbitrary shape and a finite number of C^0 points, evaluate the mechanism's kinematic performance with 3D printed prototypes, and demonstrate that the mechanism can be generalized for exact 3D motion generation.

CCS Concepts: • Computing methodologies → Shape modeling.

Additional Key Words and Phrases: cam-linkage mechanism, 3D path generation, computational design, 3D printing

ACM Reference Format:

Yingjie Cheng, Peng Song, Yukun Lu, Wen Jie Jeremy Chew, and Ligang Liu. 2022. Exact 3D Path Generation via 3D Cam-Linkage Mechanisms. *ACM Trans. Graph.* 41, 6, Article 225 (December 2022), 13 pages. <https://doi.org/10.1145/3550454.3555431>

Authors' addresses: Yingjie Cheng, University of Science and Technology of China, China, and Singapore University of Technology and Design, Singapore, chengyj@mail.ustc.edu.cn; Peng Song, Singapore University of Technology and Design, Singapore, peng_song@sutd.edu.sg; Yukun Lu, University of Science and Technology of China, China, and Singapore University of Technology and Design, Singapore, lyrik@mail.ustc.edu.cn; Wen Jie Jeremy Chew, Singapore University of Technology and Design, Singapore, jeremycjw@gmail.com; Ligang Liu, University of Science and Technology of China, China, lglu@ustc.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

0730-0301/2022/12-ART225 \$15.00

<https://doi.org/10.1145/3550454.3555431>

1 INTRODUCTION

Path generation is a fundamental problem of designing a mechanism to make a point move along a prescribed path [Russell et al. 2014]. Although this task can be achieved by robotic arms, they require coordinating motion of multiple actuators by using complex control software. Instead, path generation mechanisms can perform repetitive motion precisely driven by a *single actuator*, in which motion of the actuated part is embedded in mechanism components. This single-actuator-driven property brings several advantages including low-cost, lightweight, and easy to control, making them widely used in many applications such as machines [Roussel et al. 2018] and robotics [Mannhart et al. 2020].

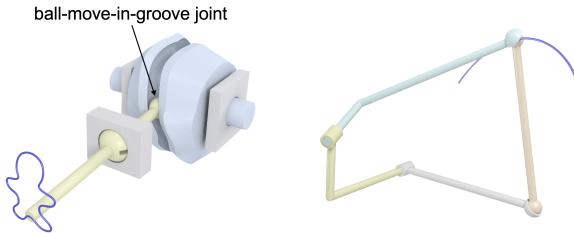


Fig. 2. Our 3D cam-linkage mechanism is a composition of (left) 3D cam-follower mechanisms [Cheng et al. 2021] and (right) a spatial linkage.

Planar linkages [Bächer et al. 2015], as well as their combinations with gears [Coros et al. 2013] and planar cams [Mundo et al. 2006] are the most widely used mechanisms for generating a path on a planar surface. Besides a planar surface, 3D cam-follower mechanisms enable to generate a path on a cylindrical or a spherical surface [Cheng et al. 2021]; see Figure 2 (left). To generate a 3D path, spatial linkages are commonly employed. However, due to the finite number of design parameters (e.g., joint positions), spatial linkages driven by a single actuator (i.e., 1-DOF spatial linkages) can only generate 3D paths with rather simple shapes [Chu and Sun 2010; Liu et al. 2020; Sun et al. 2012]; see Figure 2 (right).

In this paper, we study mechanisms that can make a point exactly move along a prescribed 3D path, a problem known as *exact 3D path generation*. Moreover, we want such mechanisms to be *simple* and *lightweight* such that they can be easily deployed into many existing systems such as machines, vehicles, or robots. As mentioned above, 1-DOF spatial linkages can only generate 3D paths with simple shapes. Extending the spatial linkage to 3-DOF enables it to exactly generate a 3D path, yet requires three independent actuators. To model an exact 3D path generation mechanism, our idea is to first model a 3-DOF spatial linkage to exactly generate a prescribed 3D path and then to reduce the spatial linkage's DOFs from 3 to 1 by composing the linkage with 3D cam-follower mechanisms since these mechanisms are able to convert 1-DOF rotation to arbitrary 2-DOF motion [Cheng et al. 2021]. We call our new mechanism *3D cam-linkage* since it is a composition of 3D cam-follower mechanisms and a spatial linkage; see Figure 1 for an example.

Given a user-specified 3D path represented as a continuous space curve with a finite number of C^0 points, our problem is to model and design a simple 3D cam-linkage mechanism that can exactly generate the path. This problem is challenging in three aspects. First, determining the 3D cam-linkage mechanism's topology is non-trivial since there are many different ways to combine 3D cam-follower mechanisms with a spatial linkage. Second, the 3D cam-linkage mechanism's end-effector should be able to exactly follow the 3D path while the whole mechanism's motion should be smooth, collision-free, and singularity-free. Third, the 3D cam-linkage mechanism should be fabricable and as lightweight as possible.

To address the challenges, we make the following contributions:

- We introduce a new 3D cam-linkage mechanism for exact 3D path generation, which only consists of two 3D cams and five links.
- We enumerate all possible topologies of the 3D cam-linkage mechanism, and present a unified method to model the geometry and kinematics for the mechanism with different topologies.

- We propose an optimization-based approach to designing a 3D cam-linkage mechanism for exactly generating a continuous 3D path while ensuring the mechanism's fabricability and minimizing its weight.

We demonstrate that our 3D cam-linkage mechanism is able to exactly generate a continuous 3D path with arbitrary shape and a finite number of C^0 points, and verify its kinematic performance with 3D printed prototypes. Although we focus on path generation in this paper, we show that our 3D cam-linkage mechanism can be generalized for accomplishing more complex motion transfer tasks such as exact rigid body guidance (i.e., exact 3D motion generation).

2 RELATED WORK

Mechanism design. The goal of mechanism design is to construct a mechanism to transfer an input motion (typically from a motor) to an output motion that satisfies a set of motion characteristics. Two typical mechanism design problems are *rigid body guidance* and *path generation*. Rigid body guidance aims to design a mechanism for guiding a rigid body through a series of specified positions and orientations. Researchers in the graphics community have addressed this problem to design various 3D printable mechanical automata, including mechanical figures that mimic human motions [Ceylan et al. 2013], planar mechanical characters [Megaro et al. 2014], kinetic wire characters [Xu et al. 2018], multi-pose mechanical objects [Nishida et al. 2019], and invertible paradoxical loop structures [Li et al. 2020]. Rather than focusing on a rigid body, path generation aims to design a mechanism to make a point move along a prescribed trajectory. Several computational approaches and tools have been developed to address the 2D path generation problem, e.g., for designing mechanical characters [Coros et al. 2013; Thomaszewski et al. 2014], drawing devices [Roussel et al. 2018], and walking machines [Bharaj et al. 2015]. When the output motion is restricted to 1-DOF translation and/or 1-DOF rotation, such as end-effector motion of mechanical toys [Zhu et al. 2012] and wind-up toys [Song et al. 2017], there is no specific difference between rigid body guidance and path generation.

2D path generation. Path generation problem can be classified as *2D* or *3D*, depending on whether the prescribed path is planar or spatial. Path generation problem also can be classified as *exact* or *approximate*, depending on whether the mechanism can exactly generate the prescribed path or not.

The most widely used mechanism for 2D path generation is 1-DOF planar linkage due to its simplicity and ease-of-fabrication. However, due to the finite number of design parameters (e.g., joint positions), planar linkages alone can only approximate a prescribed path [Bächer et al. 2015; Pan et al. 2019; Thomaszewski et al. 2014], or generate a path that exactly passes through a few prescribed 2D points (e.g., 9 precision points for 4-bar planar linkages [Wampler et al. 1992]). To address the exact 2D path generation problem, planar linkages have to be combined with mechanical parts whose shape is defined by an infinite number of parameters, in particular, planar cams [Kay and Haws 1975; Mannhart et al. 2020; Mundo et al. 2006; Singh 1981] or non-circular gears [Mundo et al. 2009]. Besides planar linkages, Cheng et al. [2021] proposed a 3D cam-follower mechanism with a ball-move-in-groove joint for exact path

generation. The mechanism can exactly generate a path not only on a planar surface, but also on a cylindrical or spherical surface, defined by the follow-support joint.

3D path generation. To generate a 3D path, the most typical way is to use 1-DOF spatial linkages in which the links can move in the 3D space driven by a single actuator. Similar to their 2D counterparts, these spatial linkages can only approximate a prescribed 3D path [Chu and Sun 2010; Liu et al. 2020; Sun et al. 2012], or generate a path that exactly passes through a few prescribed 3D points (e.g., 7 precision points for UR-2SS linkages [Chung 2015] and 9 precision points for RCCC linkages [Bai et al. 2022]). Compared with these spatial linkages, our 3D cam-linkage mechanism is able to exactly generate a prescribed continuous 3D path with arbitrary shape and a finite number of C^0 points.

One possible way to generate 3D paths with complex shapes is to combine spatial linkages with planar cams. Takahashi and Okuno [2018] developed an automaton that can draw user-specified 2D sketches with disconnected segments. The core component of the automaton is a spatial cam-linkage mechanism that combines 3 RSSR linkages with 3 planar cams to move a pen along a 3D trajectory precomputed from the user's 2D sketch. Although this cam-linkage mechanism can generate a complicated 3D path, it has a complex topology with 20 mechanical parts that are spread in the 3D space. In contrast, our 3D cam-linkage mechanism has three advantages. First, our mechanism has a simpler topology with 7 mechanical parts only (i.e., 2 3D cams and 5 links). Second, our mechanism can exactly generate a continuous 3D curve which is taken as a hard constraint (see Equation 14) in our mechanism design approach while the design approach in [Takahashi and Okuno 2018] aims to minimize the pointwise distance between the generated path and the input curve. Third, our mechanism can be generalized for accomplishing more complex motion transfer tasks such as exact 3D rigid body guidance (see Figure 15). Compared with [Takahashi and Okuno 2018], one limitation of our mechanism is fabricability. Due to the complex geometry of the 3D cams, our mechanism can be fabricated only by 3D printing with support materials while the planar cams in [Takahashi and Okuno 2018] can be fabricated by either support-free 3D printing or laser cutting.

3 PROBLEM AND OVERVIEW

Our problem is to model and design a 3D cam-linkage mechanism that can exactly generate a user-specified path, driven by a single actuator. The user-specified path is represented as a continuous space curve, either open or closed, denoted as $\mathbf{R}(u)$, $u \in [0, 1]$. The input curve $\mathbf{R}(u)$ is not necessary to be smooth; i.e., $\mathbf{R}(u)$ can have a finite number of C^0 points, denoted as $\{\mathbf{R}(u_h)\}$. In case $\mathbf{R}(u)$ is an open curve with two endpoints p_1 and p_2 , we parameterize $\mathbf{R}(u)$ in a way that $\mathbf{R}(0) = p_1$, $\mathbf{R}(0.5) = p_2$, and $\mathbf{R}(1) = p_1$. Obviously, both $\mathbf{R}(0)$ and $\mathbf{R}(0.5)$ are C^0 points in the open curve. The input curve $\mathbf{R}(u)$ can be prepared by users in different ways, e.g., creating 3D parametric curves using 3D modeling tools or projecting 2D curves onto a 3D surface; see Figure 3 for examples.

Besides exact path generation, the 3D cam-linkage mechanism should also satisfy the following requirements related to topology, fabrication, and kinematics:

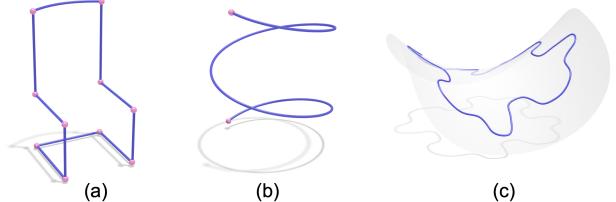


Fig. 3. Users can specify the target path in different ways: model (a) a closed or (b) an open 3D parametric curve, or (c) project a 2D curve (e.g., a jigsaw puzzle piece boundary) onto a 3D surface (e.g., a saddle surface). Note that C^0 points in (a&b) are colored in pink.

- (1) *Simple mechanism.* The mechanism should consist of as few mechanical parts as possible.
- (2) *Lightweight mechanism.* The size of each component part in the mechanism should be as small as possible, to minimize the total weight.
- (3) *Collision-free motion.* There is no collision among the mechanical parts during their motion. This also implies that each part motion should be within the range allowed by the mechanical joints.
- (4) *Singularity-free motion.* The mechanism's motion should always remain a safe distance away from singularities of the spatial linkage.
- (5) *Smooth motion.* The acceleration of the mechanical parts should be as low as possible to ensure smooth motion, assuming the input motion is a uniform rotation.

Overview of our approach. In Section 4, we study topologies of the 3D cam-linkage mechanism such that it can perform exact 3D path generation with a minimal number of mechanical parts. We propose a unified method to parameterize the geometry of the mechanism, as well as formulate analytical equations for modeling the kinematics of the mechanism. Taking the mechanism modeling as a foundation, Section 5 addresses the problem of designing 3D cam-linkage mechanisms for exact 3D path generation. We formulate the design problem as an optimization to search for an optimal topology and geometry of a 3D cam-linkage mechanism that can satisfy our design requirements. We also propose an optimization solver method that can explore the large design space efficiently.

4 MODELING 3D CAM-LINKAGE MECHANISM

This section presents our approach to modeling the 3D cam-linkage mechanism's topology, geometry, and kinematics, respectively.

4.1 Topology Modeling

Two essential components in our 3D cam-linkage mechanisms are spatial linkages and 3D cams. Spatial linkages are necessary since they are the primary mechanisms for 3D path generation. The simplest spatial linkages are 4-bar and 5-bar linkages. We choose a 5-bar spatial linkage for our mechanism, and explain why a 4-bar spatial linkage is insufficient for our goal in the supplementary material. To enable exact 3D path generation, our 5-bar spatial linkage should have 3 DOFs. Rather than controlling the linkage's 3 DOFs using 3 independent actuators, we propose to use 3D cams [Cheng et al. 2021] to control the linkage motion, aiming to reduce the whole

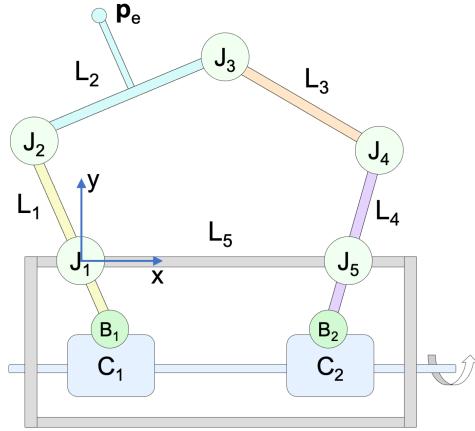


Fig. 4. An illustration of our 3D cam-linkage mechanism, which consists of five links $\{L_i\}$, five link joints $\{J_k\}$, two 3D cams $\{C_j\}$, two cam joints $\{B_j\}$, one end-effector point p_e , and a support structure (in grey).

mechanism's DOFs from 3 to 1 such that it can be driven by a single actuator; see Figure 4. Note that planar cams are unable to accomplish this task since they can only transfer 1-DOF motion to another 1-DOF motion.

Our 5-bar spatial linkage consists of five links connected by five joints, denoted as $\{L_i\}$ and $\{J_k\}$, $1 \leq i, k \leq 5$, respectively. Among the five links, L_5 is the fixed one, and the two links connected with L_5 are L_1 and L_4 . Links L_5 and L_1 are connected by joint J_1 while links L_5 and L_4 are connected by joint J_5 ; see Figure 4. We consider joints J_1 and J_5 as active joints in the 5-bar linkage while the other joints (i.e., J_2, J_3, J_4) as passive joints. Since the linkage has 3 DOFs, we assume that the active joint J_1 has 2 DOFs and the active joint J_5 has 1 DOF, without loss of generality. To reduce the mechanism's DOFs from 3 to 1, we use two 3D cams, denoted as $\{C_j\}$, $1 \leq j \leq 2$, to control the motion of the two active joints, respectively. We intentionally make the two 3D cams C_1 and C_2 co-axis such that they can be driven by a single actuator directly, without using additional mechanical parts such as gears; see Figure 4. More specifically, we use cam C_1 to control 2-DOF motion of joint J_1 and use cam C_2 to control 1-DOF motion of joint J_5 . By this, C_1, J_1 , and L_1 as well as C_2, J_5 , and L_4 form two cam-follower mechanisms denoted as $C_1 - L_1$ and $C_2 - L_4$, respectively. The cam-follower joints in $C_1 - L_1$ and $C_2 - L_4$ are denoted as B_1 and B_2 , respectively.

We call $\{B_j\}$, $1 \leq j \leq 2$, *cam joints*, and $\{J_k\}$, $1 \leq k \leq 5$, *link joints*. Each cam joint B_j is always a ball-move-in-groove joint proposed in [Cheng et al. 2021]; see Figure 2 (left). Each link joint J_k can only be a commonly used mechanical joint (i.e., R, P, C, U, and S joints) to satisfy our simple mechanism requirement. Table 1 shows the geometry, name, and allowed motion for each of these link joints. Since the active joint J_1 has 2-DOF motion while the active joint J_5 has 1-DOF motion, joint J_1 can only be {C, U} while joint J_5 can only be {R, P} according to Table 1.

We have determined candidates for the two active joints J_1 and J_5 in the 5-bar linkage. However, there are still three passive joints (i.e., J_2, J_3 , and J_4) remaining in the linkage. The DOFs f_k of each joint J_k in the 5-bar linkage should satisfy the mobility formula [Uicker

Table 1. Link joints used in our 3D cam-linkage mechanism and their allowed degrees of freedom s_k , among which α, β , and γ are the rotation angles while d is the translation distance. Note that \hat{d} is the maximum translation distance allowed by P and C joints.

	Joint name	Abbrev.	Motion	v_k	s_k	Motion Range
	Revolute joint	R	1-DOF Rotation	Rotation axis	α	$\alpha \in [-180^\circ, 180^\circ]$
	Prismatic joint	P	1-DOF Translation	Translation axis	d	$d \in [0, \hat{d}]$
	Cylindrical joint	C	1-DOF Rotation + 1-DOF Translation	Rotation & translation axis	$[\alpha, d]$	$\alpha \in [-180^\circ, 180^\circ]$ $d \in [0, \hat{d}]$
	Universal joint	U	2-DOF Rotation	Axis of not allowed 1-DOF rotation	$[\alpha, \beta]$	$\alpha \in [-45^\circ, 45^\circ]$ $\beta \in [-180^\circ, 180^\circ]$
	Spherical joint	S	3-DOF Rotation	-	$[\alpha, \beta, \gamma]$	$\alpha \in [-45^\circ, 45^\circ]$ $\beta \in [-180^\circ, 180^\circ]$ $\gamma \in [-45^\circ, 45^\circ]$

et al. 2016]:

$$M = 6(N - 1 - K) + \sum_{k=1}^K f_k \quad (1)$$

where $M = 3$ is the linkage's DOFs, $N = 5$ is the number of links, and $K = 5$ is the number of link joints. Recall that $f_1 = 2$ and $f_5 = 1$. Substituting the values of M, N, K, f_1 and f_5 into Equation 1, we have:

$$f_2 + f_3 + f_4 = 6 \quad (2)$$

We enumerate all possible combinations of link joints in Table 1 that satisfy Equation 2, and find that the total number of combinations is 128; please refer to the supplementary material for details. Each combination of the link joints (e.g., $J_1J_2J_3J_4J_5 = \text{URSUR}$) defines a unique topology of the 3D cam-linkage mechanism. Hence, we have 128 different topologies of the 3D cam-linkage mechanism. To compare these topologies, we conducted an experiment and found that P and C joints typically result in L-shaped links and a larger motion envelope of the linkage, which violates our requirement on a lightweight mechanism; see Section 6 and Figure 9. Hence, we only use mechanism topologies without P or C joints for exact 3D path generation in this paper. There are 7 such topologies in total, i.e., $\text{URSUR}, \text{URUSR}, \text{USRUR}, \text{USURR}, \text{UUSR}, \text{UURSR}, \text{UUUR}$; see Figure 5 for examples.

4.2 Geometry Modeling

Our 3D cam-linkage mechanism consists of five different kinds of components. Below, we describe our approach to modeling the geometry for each kind of component:

i) *Link joints*. Table 1 shows five different types of link joints used in our mechanism, i.e., R, P, C, U, and S joints, as well as their geometry and allowed degrees of freedom s_k . Note that the U joint is actually a modified S joint that removes 1-DOF rotation by inserting two pins in the S joint. Each of the R, P, C, and U joints is associated with a 3D axis, denoted as v_k ; see Table 1 for the meaning of v_k . The geometry of each link joint defines the allowed range of relative motion between the two associated links. Due to the fabrication

constraint, both U and S joints can support $[-180^\circ, 180^\circ]$ rotation around the spinning axis and $[-45^\circ, 45^\circ]$ rotation for each of the other rotation axes. The maximum translation distance \hat{d} allowed by the P and C joints is defined by the length of the joint and the associated link.

ii) Cam joints. Each cam joint is modeled as a ball-move-in-groove joint, which allows 4-DOF motion of the follower relative to the cam (i.e., 3-DOF rotation and 1-DOF translation along the cam groove); see Figure 2 (left). The joint geometry is parameterized by the 3D cam's pitch curve denoted as Q_j and the follower ball radius r [Cheng et al. 2021].

iii) Links. Among the five links, L_5 is a fixed link; L_1 and L_4 are driven by the active joints J_1 and J_5 respectively, called *active links*; and L_2 and L_3 are driven by the passive joints J_2 , J_3 , and J_4 , called *passive links*. Each active/passive link is modeled as a bar (with a circular cross-section) that connects two neighboring link joints (e.g., L_2 connects J_2 and J_3) or two neighboring link joints as well as a cam joint (e.g., L_1 connects J_1 , J_2 , and B_1); see Figure 4. The shape of each link is not necessary to be always a straight bar. To avoid collision when transferring motion, we model the centerline of each passive link as a polyline with three line segments, among which the middle segment is usually the longest one. The centerline of each active link has three more line segments for connecting the cam joint; see Figure 5 for examples.

iv) End-effector. To output 3-DOF motion, the end-effector has to be put on the passive link L_2 or L_3 since the active links L_1 and L_4 can only output 2-DOF and 1-DOF motion, respectively. Without loss of generality, we always put the end-effector on the passive link L_2 . The geometry of the end-effector is modeled as a bar extended from the link L_2 . The point on the end-effector for generating a path is called the *end-effector point* and denoted as p_e ; see Figure 4.

v) 3D cams. The 3D cam geometry is modeled as a watertight surface that holds the cam joint following the approach in [Cheng et al. 2021]. One modification that we make is to remove some unnecessary volume of the 3D cam to make it more lightweight; compare Figures 5 and 2.

vi) Support. The support of our 3D cam-linkage mechanism is a structure that holds the common rotation axis of the two 3D cams as well as the fixed link L_5 . We model the support structure in two different ways: a baseboard with three supporting pillars for visualizing the mechanism (see Figure 1) or a box frame that encloses the 3D cams for fabricating the mechanism (see Figure 12).

4.3 Kinematic Modeling

In this section, we first describe our approach to modeling the kinematics of the 5-bar spatial linkage, considering joints J_1 and J_5 as active joints; see Section 4.3.1. In Section 4.3.2, we model the kinematics of the 3D cam-linkage mechanism by considering it as a composition of the 5-bar spatial linkage (i.e., $L_1L_2L_3L_4L_5$) and two cam-follower mechanisms (i.e., $C_1 - L_1$, $C_2 - L_4$).

We define a local coordinate system for our 3D cam-linkage mechanism, where the origin is at the center of joint J_1 , the x -axis points

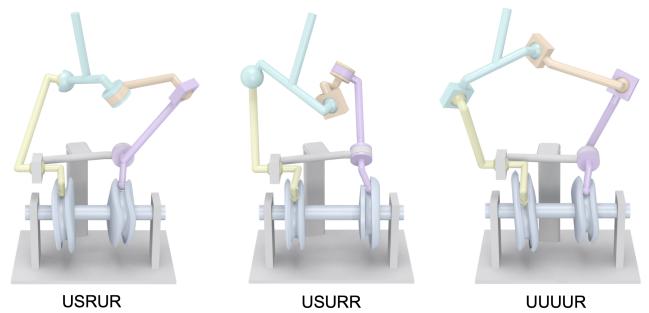


Fig. 5. Three example topologies of our 3D cam-linkage mechanism.

toward the center of joint J_5 , and the y -axis points toward the center of joint J_2 (and is perpendicular to x -axis) in the mechanism's initial configuration; see Figure 4. Variables for kinematic modeling of the mechanism are all defined in this coordinate system, unless otherwise specified.

4.3.1 Kinematics of 5-bar Spatial Linkage. We model both forward kinematics (FK) and inverse kinematics (IK) of the 5-bar spatial linkage, and perform singularity analysis.

Forward kinematics. Forward kinematics of a linkage aims to determine the linkage's state for given actuation parameters. The state s of a linkage is defined through its time-varying joint angles, i.e., $s = [s_1(t), s_2(t), \dots, s_5(t)]$; see Table 1 for details. Note that we consider the translation distance d of the P and C joints as a generalized joint angle, to simplify the terminology. The actuation parameters of our 5-bar spatial linkage are the angles of active joints J_1 and J_5 , i.e., s_1 and s_5 . The forward kinematics of the linkage can be expressed as the following motion transfer function:

$$[s_2, s_3, s_4] = f_{\text{linkage}}([s_1, s_5]) \quad (3)$$

Assuming the linkage's initial state $\bar{s} = s(0)$ is known and valid, the constraint on the state s at time t enforced by the single loop in our 5-bar spatial linkage can be formulated as (see also Figure 6):

$$C_L(s_1, s_2, s_3, s_4, s_5) = 0 \quad (4)$$

where C_L is a column vector with 12 elements. Please refer to the appendix for details about computing the constraint C_L . Given the known actuation parameters s_1 and s_5 , we can then solve for the unknown passive joint angles $[s_2, s_3, s_4]$ of the linkage by minimizing a non-linear least squares problem:

$$\min_s C_L^T C_L \quad (5)$$

We solve the optimization problem using a gradient-based method, e.g., Levenberg-Marquardt algorithm.

Once we compute the kinematic equation $f_{\text{linkage}}(\cdot)$, the end-effector point position $p_e(t)$ on link L_2 at time t can be calculated using:

$$p_e(t) = M_{L_2}(t) \bar{p}_e, \quad t \in [0, T] \quad (6)$$

where $M_{L_2}(t)$ is the transformation of link L_2 at time t (see the appendix for details), $M_{L_2}(0) = I$, and $\bar{p}_e = p_e(0)$. When the active joints J_1 and J_5 move for a whole period T , the trajectory of the

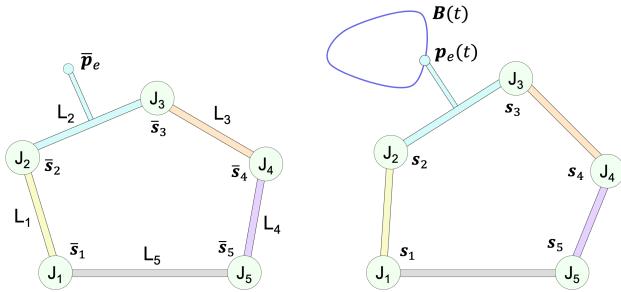


Fig. 6. Our 5-bar spatial linkage in (left) the initial state at time $t = 0$ and (right) the current state at time t .

end-effector point $p_e(t)$ is the path generated by the 5-bar linkage denoted as $B(t)$; see Figure 6 (right).

Inverse kinematics. The goal of inverse kinematics is to determine the linkage's state s (i.e., both passive and active joint angles) that can generate each point on a given path $R(u)$, $u \in [0, 1]$. Starting from the known initial state \bar{s} , we determine a unique state $s(t)$ iteratively by formulating the following optimization problem:

$$\begin{aligned} \min_s \quad & \|s(t) - s(t - \Delta t)\| \\ \text{s.t.} \quad & C_L^T C_L = 0, \\ & M_{L_2}(t) \bar{p}_e = R(u) \end{aligned} \quad (7)$$

where Δt is a small time duration in the motion period T ($\Delta t = T/1000$ in our experiments). Note that a valid solution s can always be found by our optimization as long as the path $R(u)$ is in the reachable space of the linkage's end-effector, since our 5-bar linkage has 3 DOFs. In practice, we usually do not know whether a given path $R(u)$ is in the reachable space of the linkage's end-effector or not. To handle this case, we use a gradient-based method to solve the dual problem instead:

$$\min_s \quad \|C_L\|^2 + \|M_{L_2} \bar{p}_e - R\|^2 \quad (8)$$

Starting from the initial state \bar{s} , we set $s(t) = s(t - \Delta t)$, for $t \in [0, T]$, as the initial value of $s(t)$, and then use a gradient-based method to find the optimal value of $s(t)$. In case that our method fails to find a solution for Equation 8, it means that the 5-bar linkage cannot reach the point $R(u)$.

Singularity analysis. We analyze FK singularities of the linkage following the approach in [Gosselin and Angeles 1990]. Differentiating Equation 4 with respect to time t leads to the relationship between the input and output speeds as follows:

$$A [\dot{s}_2, \dot{s}_3, \dot{s}_4] + B [\dot{s}_1, \dot{s}_5] = 0$$

where $A = \frac{\partial C_L}{\partial [s_2, s_3, s_4]}$ is the FK Jacobian matrix. The minimal singular value $\sigma_{\min}(A)$ represents the distance from the linkage state $s(t)$ to the FK singularities.

We define the IK Jacobian matrix as $J = \frac{\partial C_{IK}}{\partial [s_1, s_2, s_3, s_4, s_5]}$, where $C_{IK} = [C_L, (M_{L_2} \bar{p}_e - R)]$ according to Equation 7. Condition number $\kappa(J)$ describes the distance from the linkage state $s(t)$ to the IK singularities [Merlet 2006; Yoshikawa 1985]. Based on the implicit function theorem, $\kappa(J)$ quantifies smoothness of the linkage motion.

4.3.2 Kinematics of Cam-Linkage Mechanism. In our 3D cam-linkage mechanism, the two active joints angles s_1 and s_5 of the 5-bar linkage is actually the output motion of the two cam-follower mechanisms $C_1 - L_1$ and $C_2 - L_4$, respectively. Hence, the 3D cam-linkage mechanism's forward kinematics can be obtained by composing those of the 5-bar linkage and the two cam-follower mechanisms.

The motion transfer equation of the two cam-follower mechanisms can be written as:

$$s_1 = f_{cam1}(\theta), \quad (9)$$

$$s_5 = f_{cam2}(\theta) \quad (10)$$

where θ is the rotation angle of the single actuator. We compute the two motion transfer functions $f_{cam1}(\cdot)$ and $f_{cam2}(\cdot)$ following the approach in [Cheng et al. 2021]. Combing Equations 3, 9, and 10, we obtain the motion transfer function of our 3D cam-linkage mechanism:

$$[s_1, s_2, s_3, s_4, s_5] = f_{mech}(\theta) \quad (11)$$

where $f_{mech}(\cdot)$ is a composition function of $f_{linkage}(\cdot)$, $f_{cam1}(\cdot)$, and $f_{cam2}(\cdot)$.

5 DESIGNING 3D CAM-LINKAGE MECHANISM

This section presents our approach to designing 3D cam-linkage mechanisms that satisfy the high-level requirements in Section 3. We formulate the design problem as an optimization in Section 5.1 and then present our method to solve the optimization in Section 5.2.

5.1 Optimization Formulation

Taking a user-specified continuous space curve $R(u)$, $u \in [0, 1]$ defined in the world space as input, we aim to design a 3D cam-linkage mechanism whose end-effector point p_e can generate a path that is exactly the same as the curve $R(u)$. We assume that the position and orientation of the mechanism in the world space have been specified by users, represented as a matrix T_{mech} .

Search space. Our search space includes both the topology and geometry of the 3D cam-linkage mechanism. The topology search space is the 7 topologies introduced in Section 4.1, denoted as $\{G_i\}$. The geometry search space is the parameters that define the 3D cam-linkage mechanism's geometry in the mechanism's local coordinate system. We classify these parameters into two classes: *kinematic-related parameters* that determine motion of the mechanism, and *fabrication-related parameters* that affect only fabrication but not motion of the mechanism. The kinematic-related parameters, denoted as Ψ include:

- (1) $\{x_k(0)\}$, $1 \leq k \leq 5$ that defines the position and orientation of each link joint J_k in the initial state of the mechanism at $t = 0$.
- (2) $\{Q_j(w)\}$, $w \in [0, 1]$, $1 \leq j \leq 2$ that defines the pitch curve of each 3D cam C_j . $Q_j(w)$ is modeled as a cubic spline where $Q_j(0)$ represents the position of the cam joint B_j at $t = 0$.

The fabrication-related parameters include geometric parameters of each link joint (e.g., radius of S and U joints), follower ball radius r in each cam joint, centerline of each link, radius of each link's circular cross-section, as well as geometric parameters of the support structure. The fabrication-related parameters are either predefined or

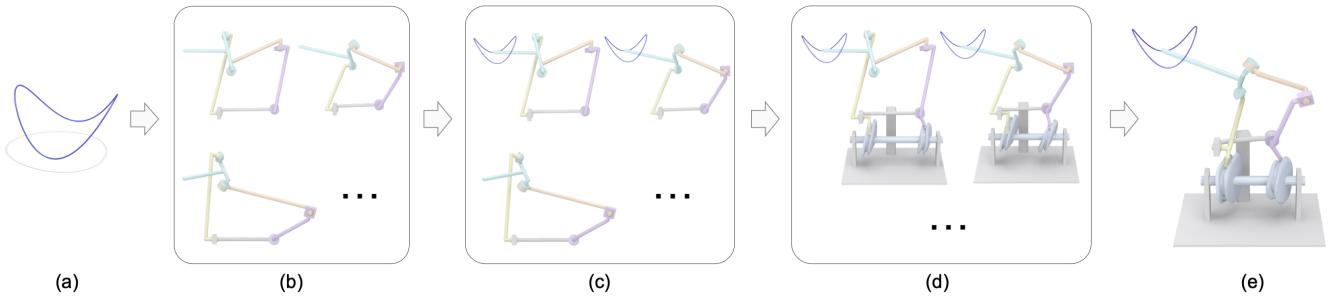


Fig. 7. Our approach to solving the 3D cam-linkage mechanism design problem, assuming a fixed topology. (a) Given a user-specified 3D curve, we first (b) initialize a set of 5-bar linkage candidates $\{L_i\}$ and then (c) compute the inverse kinematics for each linkage L_i to determine the linkage state $s_i(t)$. (d) For each linkage candidate with valid IK, we compute the geometry Q_1 and Q_2 of the two 3D cams based on the two joint angles $s_1(t)$ and $s_5(t)$, respectively. (e) Our output is an optimal cam-linkage design that satisfies all the constraints and minimizes the objective function in Section 5.1.

calculated as post-processing after the kinematic-related parameters have been determined.

To quantify a mechanism design, we have to simulate the mechanism's forward kinematics for a whole motion period T . Hence, we also need to determine the joint angle parameters $s(t) = [s_1(t), s_2(t), s_3(t), s_4(t), s_5(t)]$, $t \in [0, T]$ by computing the motion transfer function in Equation 11. Without loss of generality, we assume the actuator rotates with a constant angular speed ω at any time t , i.e., $\theta(t) = \omega t$, when simulating the mechanism's kinematics.

Objective function. Our mechanism design problem is formulated as minimizing the following objective function:

$$E(\Psi) = \lambda_1 E_{\text{smth}}(\Psi) + \lambda_2 E_{\text{weig}}(\Psi) + \lambda_3 E_{\text{link}}(\Psi) \quad (12)$$

with $E_{\text{smth}} = \max_{t \in [0, T]} \kappa(J)$,

$$E_{\text{weig}} = \sum_{j=1}^2 \text{Len}(Q_j(w)) + \sum_{i=1}^5 \text{Len}(L_i),$$

$$\text{and } E_{\text{link}} = \sqrt{\sum_{i=1}^5 (\text{Len}(L_i) - \bar{L})^2},$$

where E_{smth} measures the smoothness of the linkage motion, E_{weig} approximates the mechanism's weight based on the length of each pitch curve $Q_j(w)$ and the length of each link L_i , E_{link} is the standard deviation of the five links' lengths, \bar{L} is the average length of the five links in the linkage, and λ_1 , λ_2 , and λ_3 are weights of the three energy terms. We set $\lambda_1 = 5$, $\lambda_2 = 1$, and $\lambda_3 = 10$ in our experiments to balance the impact of the three energy terms.

As mentioned in Section 4.2, there are two kinds of links: $\{L_2, L_3, L_5\}$ that connect two link joints, and $\{L_1, L_4\}$ that connect two link joints and one cam joint. Denote \tilde{x}_k as the positional component of $x_k(0)$. We approximate the length of each link L_i in $\{L_2, L_3, L_5\}$ as $\|\tilde{x}_a - \tilde{x}_b\|$, where \tilde{x}_a and \tilde{x}_b are the positions of the two joints connected by link L_i . We approximate the length of each link L_i in $\{L_1, L_4\}$ as $\|\tilde{x}_a - \tilde{x}_b\| + \|\tilde{x}_b - Q_j(0)\|$.

Constraints. We formulate various constraints related to kinematics and fabrication as follows.

- (1) *Linkage kinematics.* According to Equation 4, the 5-bar linkage should satisfy the following constraint to ensure validity of its

kinematics:

$$C(s(t))_L^T C(s(t))_L = 0, \quad t \in [0, T] \quad (13)$$

- (2) *Exact path generation.* The end-effector's trajectory $B(t)$ should be exactly the same as the input curve $R(u)$:

$$T_{\text{mech}} B(t) = R(u), \quad t \in [0, T], u \in [0, 1] \quad (14)$$

where the left-hand side is the actual trajectory of the end-effector point p_e in the world space.

For each C^0 point $R(u_h)$ in the input curve, the end-effector's speed when passing the point has to be zero. To this end, the two active joint angles s_1 and s_5 that drive the 5-bar linkage should have zero gradient (i.e., zero angular speed) at this moment:

$$\begin{aligned} f_{\text{cam}1}(\theta_h) &= s_{1,h}, \quad f'_{\text{cam}1}(\theta_h) = \mathbf{0} \\ f_{\text{cam}2}(\theta_h) &= s_{5,h}, \quad f'_{\text{cam}2}(\theta_h) = \mathbf{0} \end{aligned} \quad (15)$$

where $s_{1,h}$, $s_{5,h}$, and θ_h are the joint angle s_1 , joint angle s_5 , and actuator rotation angle θ when the end-effector passes the C^0 point $R(u_h)$, respectively.

- (3) *Singularity-free motion.* The minimal distance of the mechanism to FK singularities during a whole motion period T should be larger than a threshold μ :

$$\min_{t \in [0, T]} (\sigma_{\min}(A)) \geq \mu, \quad (16)$$

where $\mu = 0.01$ in our experiments.

- (4) *Collision-free motion.* To ensure collision-free motion, we require that link joints should not be too close to one another during any time t :

$$Dist(\tilde{x}_k(t), \tilde{x}_l(t)) > \tau, \quad 1 \leq k < l \leq 5, \quad t \in [0, T] \quad (17)$$

where τ is the allowed minimal distance between any pair of the link joints. We set $\tau = 6r$ empirically in our experiments, where r is the radius of the follower ball in the cam joint. Thanks to the distance constraints on the link joints, link-link collision can be avoided by varying the links' shape as a post-process.

We avoid collision between the linkage (i.e., passive links) and the fixed link L_5 (i.e., support) by requiring:

$$\tilde{x}_k^y(t) > \tau/2, \quad 2 \leq k \leq 4, \quad t \in [0, T] \quad (18)$$

where $\tilde{x}_k^y(t)$ is the y-coordinate of $\tilde{x}_k(t)$ in the mechanism's local coordinate system. We avoid collision between the two 3D cams by requiring:

$$\text{dist}(\mathbf{Q}_1, \mathbf{Q}_2) > \tau \quad (19)$$

We avoid collision between each 3D cam and the fixed link L_5 by requiring:

$$\tilde{x}_5^y - (\mathbf{M}_R(\theta(t)) \mathbf{Q}_j(w))^y > \tau/2, \quad 1 \leq j \leq 2, \quad t \in [0, T], \quad w \in [0, 1] \quad (20)$$

where $\mathbf{M}_R(\theta(t))$ is the actuator's rotation matrix at time t .

- (5) *Link joint angles.* The joint angle of each link joint should be within a valid range:

$$\mathbf{s}_k(t) \in \Omega_l, \quad 1 \leq k \leq 5, \quad t \in [0, T] \quad (21)$$

where Ω_l is the set of valid joint angles defined for the joint type l in Table 1.

- (6) *Smooth cam groove.* The pitch curve $\mathbf{Q}_j(w)$ should be closed and smooth such that the resulting cam groove allows the follower ball to move periodically and smoothly in it:

$$\rho_j(w) > \zeta r, \quad 1 \leq j \leq 2, \quad w \in [0, 1] \quad (22)$$

where $\rho_j(w)$ is the radius of curvature of $\mathbf{Q}_j(w)$, r is the follower ball radius, and ζ is a coefficient typically set as 1.2 in our experiments.

5.2 Optimization Solver

Our optimization problem in Section 5.1 is challenging to solve. First, the search space is large, including mechanism topologies $\{G_l\}$, linkage parameters $\{\mathbf{x}_k(0)\}$, and cam pitch curves $\{\mathbf{Q}_j\}$. Second, there are a large number of constraints, some of which are non-convex such as collision-free motion and range of link joint angles.

To address the above challenges, our observation is that the search space is dominated by the cam pitch curves $\{\mathbf{Q}_j\}$ while the mechanism topologies $\{G_l\}$ and linkage parameters $\{\mathbf{x}_k(0)\}$ have a relatively low dimension. Hence, our idea is to divide our problem into two sub-problems by considering our 3D cam-linkage mechanism as a composition of two 3D cam-follower mechanisms and a 3-DOF 5-bar linkage; see Figure 7. The first sub-problem is to compute the pitch curves $\{\mathbf{Q}_j\}$ while fixing the mechanism topology G_l and linkage parameters $\{\mathbf{x}_k(0)\}$, which can be addressed by optimizing the two 3D cam-follower mechanisms using the method in [Cheng et al. 2021]. The second sub-problem is to explore different mechanism topologies G_l and linkage parameters $\{\mathbf{x}_k(0)\}$. Since the search space is relatively small, we can enumerate all possible mechanism topologies $\{G_l\}$ and explore the linkage parameters $\{\mathbf{x}_k(0)\}$ by using an evolutionary algorithm.

In detail, for each given topology G_l , we first use the Particle Swarm Optimization (PSO) [Kennedy and Eberhart 1995] to randomly initialize N ($N = 40$ in our experiments) particles, where each particle represents the kinematic-related geometric parameters Ψ^i of a 3D cam-linkage mechanism \mathbf{M}^i . In particular, we only initialize the linkage parameters $\mathbf{x}^i = \{\mathbf{x}_k^i(0)\}$ for each particle Ψ^i and the corresponding pitch curves $\{\mathbf{Q}_j^i\}$ are computed by optimizing the two 3D cam-follower mechanisms. In detail, we first solve the inverse kinematics problem in Equation 8 for the 3-DOF spatial linkage to obtain the joint angles $\mathbf{s}^i(t) = [\mathbf{s}_1^i(t), \mathbf{s}_2^i(t), \mathbf{s}_3^i(t), \mathbf{s}_4^i(t), \mathbf{s}_5^i(t)]$, where

Algorithm 1 Algorithm to solve the 3D cam-linkage mechanism design problem, taking a user-specified curve $\mathbf{R}(u)$ as an input.

```

1: function DESIGNMECHANISM( $\mathbf{R}(u)$ )
2:    $G^{\text{optim}} \leftarrow \emptyset$ 
3:   for each topology  $G_l$  do ▷ Initialize  $N$  particles
4:      $\Psi_{\text{list}} \leftarrow \emptyset$ 
5:     for  $i = 0; i < N; i++$  do
6:        $\mathbf{x}^i = \{\mathbf{x}_k^i(0)\} \leftarrow$  initialize linkage parameters with
      topology  $G_l$ 
7:        $\mathbf{Q}_1^i \leftarrow \emptyset$ 
8:        $\mathbf{Q}_2^i \leftarrow \emptyset$ 
9:        $\Psi^i \leftarrow \{\mathbf{x}^i, \mathbf{Q}_1^i, \mathbf{Q}_2^i\}$ 
10:       $\Psi_{\text{list}}.\text{push\_back}(\Psi^i)$ 
11:       $\Psi^{\text{optim}} \leftarrow \emptyset$ 
12:      for  $\text{itr} = 0; \text{itr} < \text{itr}_{\text{max}}; \text{itr}++$  do ▷ Evaluate  $N$  particles
13:        for each particle  $\Psi^i \in \Psi_{\text{list}}$  do
14:           $\mathbf{s}^i(t) \leftarrow$  solve IK in Eq. 8 with  $\mathbf{R}(u)$  as the input
15:           $\mathbf{r}^i.\text{push\_back}(\text{residuals of Eq. 13 and 14})$ 
16:           $\mathbf{Q}_1^i \leftarrow$  optimize cam-follower  $C_1 - L_1$  to minimize
       $\text{Len}(\mathbf{Q}_1^i)$  subject to the follower motion  $\mathbf{s}_1^i(t)$ , Eq. 15, 20 and 22.
17:           $\mathbf{Q}_2^i \leftarrow$  optimize cam-follower  $C_2 - L_4$  to minimize
       $\text{Len}(\mathbf{Q}_2^i)$  subject to the follower motion  $\mathbf{s}_5^i(t)$ , Eq. 15, 20 and 22.
18:           $\mathbf{r}^i.\text{push\_back}(\text{residuals of Eq. 16, 17, 18, 19, 21})$ 
19:           $N(\Psi^i) \leftarrow \|\mathbf{r}^i\|_0$ 
20:           $S(\Psi^i) \leftarrow \|\mathbf{r}^i\|_1$ 
21:           $E(\Psi^i) \leftarrow$  evaluate  $\Psi^i$  using Eq. 12
22:           $[G^{\text{optim}}, \Psi^{\text{optim}}, N_{\text{min}}, S_{\text{min}}, E_{\text{min}}] \leftarrow$  record the
      best candidate and its three metrics ▷ Move  $N$  particles
23:        for each particle  $\Psi^i \in \Psi_{\text{list}}$  do
24:          Update linkage parameters  $\mathbf{x}^i$  in  $\Psi^i$ 
25:        Construct a mechanism  $\mathbf{M}^{\text{optim}}$  using  $[G^{\text{optim}}, \Psi^{\text{optim}}]$ 
return  $\mathbf{M}^{\text{optim}}$ 
```

$t \in [0, T]$. Since the active joint angles $[\mathbf{s}_1^i(t), \mathbf{s}_5^i(t)]$ are actually the output motions of the two cam-follower mechanisms $C_1 - L_1$ and $C_2 - L_4$ respectively, we compute the pitch curve \mathbf{Q}_1^i (\mathbf{Q}_2^i) of 3D cam C_1 (C_2) by optimizing $C_1 - L_1$ ($C_2 - L_4$) independently using the method in [Cheng et al. 2021]. By this, we obtain all the parameters $\Psi^i = \{\mathbf{x}^i, \mathbf{Q}_1^i, \mathbf{Q}_2^i\}$ of the cam-linkage mechanism \mathbf{M}^i .

To evaluate each particle Ψ^i , we check whether it satisfies the constraints listed in Section 5.1, and compute the corresponding residual \mathbf{r}^i , where $\mathbf{r}_k^i = \mathbf{0}$ if the k -th constraint in Section 5.1 is satisfied by the particle Ψ^i . For constraints related to motion, we compute the residual by discretizing the whole motion cycle T into 60 samples. We evaluate the particle Ψ^i using three metrics: the number of unsatisfied constraints ¹ $N(\Psi^i) = \|\mathbf{r}^i\|_0$, the sum of

¹A constraint is considered as being satisfied if the residual is less than an epsilon 10^{-8} .

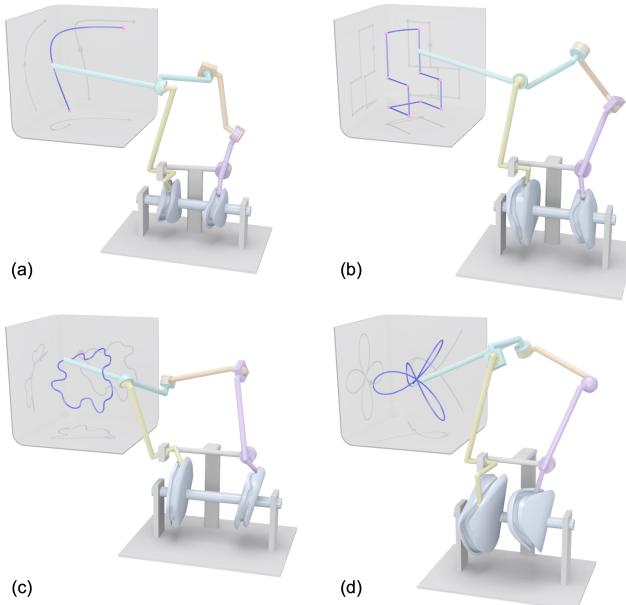


Fig. 8. Our 3D cam-linkage mechanism exactly generates a 3D path with a variety of shapes: (a) BÉZIER CURVE, (b) CHAIR, (c) PUZZLE PIECE, and (d) CONICAL ROSE.

residual of the constraints $S(\Psi^i) = \|\mathbf{r}^i\|_1$, and the objective function $E(\Psi^i)$ in Equation 12. We consider a particle Ψ^i is better than another one Ψ^j if one of the following condition is satisfied:

- (1) $N(\Psi^i) < N(\Psi^j)$
- (2) $N(\Psi^i) = N(\Psi^j) > 0$ & $S(\Psi^i) < S(\Psi^j)$
- (3) $N(\Psi^i) = N(\Psi^j) = 0$ & $E(\Psi^i) < E(\Psi^j)$

We choose the above criteria since initially many particles do not satisfy all the constraints in Section 5.1 and we want to identify and keep particles that satisfy more constraints.

To move the particles in the search space, we set the learning rate to 1 and the acceleration coefficient to 2. The inertia coefficient is linearly decreased from 0.9 to 0.4. We repeat the above PSO algorithm for each topology in $\{G_l\}$ to determine the optimal parameters $[G^{\text{optimal}}, \Psi^{\text{optimal}}]$, which will be used to construct the optimal mechanism M^{optimal} ; see Algorithm 1 for details.

6 RESULTS

We implemented our tool in C++ and libigl [Jacobson et al. 2018] on a desktop computer with a 3.7GHz CPU and 16GB memory. Figure 8 shows that our 3D cam-linkage mechanism is able to exactly generate a wide variety of curves, including an open BÉZIER curve with two C^0 points (i.e., two endpoints), a closed CHAIR curve with ten C^0 points, a PUZZLE PIECE curve projected onto a saddle surface, and a famous parametric curve CONICAL ROSE. Table 2 provides statistics of all the results shown in the paper, including the mechanism topology, values of the three energy terms, and time taken to optimize each mechanism. Please refer to the accompanying video for demos of the mechanisms and the supplementary material for equations of the input parametric curves.

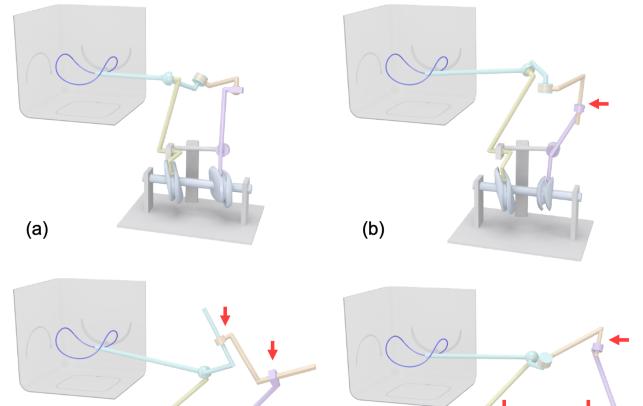


Fig. 9. Evaluating different topologies of our 3D cam-linkage mechanism with (a) 0, (b) 1, (c) 2, or (d) 3 P or C joint(s), highlighted with red arrows.

Evaluation of mechanism topologies. Since there are 128 valid topologies for our 3D cam-linkage mechanism, we want to know which topologies can lead to a more usable mechanism. Our conjecture is that 3D cam-linkage mechanisms with fewer P or C joints would be more usable since translational motion allowed by P and C joints makes the link with L-like shape, significantly increasing the linkage's motion envelope. To verify this conjecture, we conducted an experiment to compare 3D cam-linkage mechanisms with different numbers of P or C joint(s). Taking the same curve as an input, we run our design algorithm to generate the path while restricting

Table 2. Statistics of the results shown in this paper.

Fig	Curve	Topology	E_{smth}	E_{weig}	E_{link}	Optim. Topology	Optim. Time (min)	
1	Trefoil Knot	USRUR	12.9	522.7	8.2	Yes	47.8	
7	Pancake	USRUR	14.3	475.6	2.6	Yes	30.3	
8	Bézier curve	USRUR	12.5	409.3	10.7	Yes	37.9	
	Chair	USRUR	14.6	546.3	5.2	Yes	42.7	
	Puzzle Piece	USRUR	14.9	583.0	2.8	Yes	53.6	
	Conical Rose	UURSR	15.6	617.6	14.9	Yes	57.8	
9	Hyperbolic Boundary	USRUR	14.3	393.1	9.9	No	14.0	
	Hyperbolic Boundary	USRCR	19.6	458.8	13.9	No	12.7	
	Hyperbolic Boundary	USCPR	18.1	535.5	11.4	No	14.6	
	Hyperbolic Boundary	CSRPC	44.8	784.0	18.9	No	20.4	
10	Arbitrary Curve	11 C^0 Points	USRUR	12.9	862.5	7.1	No	13.2
11		2 C^0 Points	USRUR	15.8	580.3	1.8	Yes	34.5
		7 C^0 Points	USURR	18.5	672.8	2.1	Yes	47.7
		12 C^0 Points	USRUR	13.1	1441.9	8.9	Yes	52.5
		17 C^0 Points	USRUR	15.9	1672.9	16.3	Yes	49.0
12	Tennis Ball Seam	URSUR	12.6	483.6	2.7	Yes	46.4	
13	Heart	URSUR	16.2	463.9	1.4	Yes	39.9	
14	Manipulation Path	USRUR	15.7	508.1	8.3	Yes	44.9	

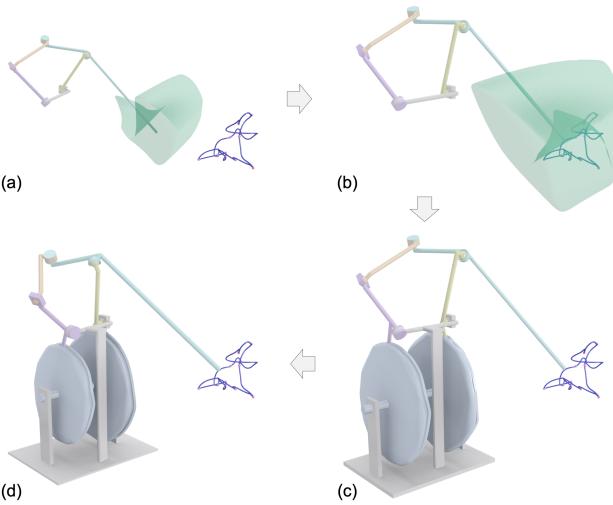


Fig. 10. Generating a 3D path with arbitrary shape and a finite number of C^0 points. (a) Compute a continuous and volumetric portion of a 3-DOF 5-bar spatial linkage's output motion space that is collision-free and singularity-free; (b) transform the linkage such that its output motion space fully covers the input curve; (c) compute the geometry of the two 3D cams; and (d) optimize the geometry of the 3D cam-linkage mechanism.

the topology to have 0, 1, 2, and 3 P or C joint(s), respectively. Figure 9(a-d) shows that the more P or C joints in the 5-bar linkage, the larger the motion envelope of the linkage, where the motion envelope size is in the ratio of 1 : 1.181 : 1.208 : 1.793.

Generating a 3D path with arbitrary shape. In this experiment, we demonstrate that our 3D cam-linkage mechanism is able to exactly generate a continuous 3D curve with arbitrary shape and a finite number of C^0 points. The exactness of 3D path generation means that the distance between each sample point (1000 sample points in our experiments) on the generated path and the target path represented as a parametric curve is less than a floating point epsilon (equals 10^{-8} in our experiments). Without loss of generality, we choose a 3-DOF 5-bar spatial linkage with topology USRUR. We choose the linkage's geometry in a way that there is a continuous and volumetric portion Φ of the linkage's output motion space that is singularity-free and collision-free, where the volumetric portion is computed using a sampling-based approach; see Figure 10(a). Next, we model a 3D curve $R(u)$ with arbitrary shape by moving a particle with random acceleration in a fixed sphere, where C^0 points are introduced in the curve by changing the particle's current velocity (integrated from the acceleration) to a random velocity with different direction. We transform (i.e., translate, rotate, and scale) the 5-bar linkage such that its motion space Φ can fully cover the curve $R(u)$; see Figure 10(b). By this, all the design constraints related to the spatial linkage (i.e., Equations 13, 14, 16, 17, 18, 21) have been satisfied. The geometry of each 3D cam C_1 (C_2) can be further generated according to the active joint angle $s_1(t)$ ($s_5(t)$) calculated from the linkage's inverse kinematics; see Figure 10(c). According to [Cheng et al. 2021], we can always find such a 3D cam that satisfies the constraints of Equations 15, 20, and 22. To ensure the collision-free

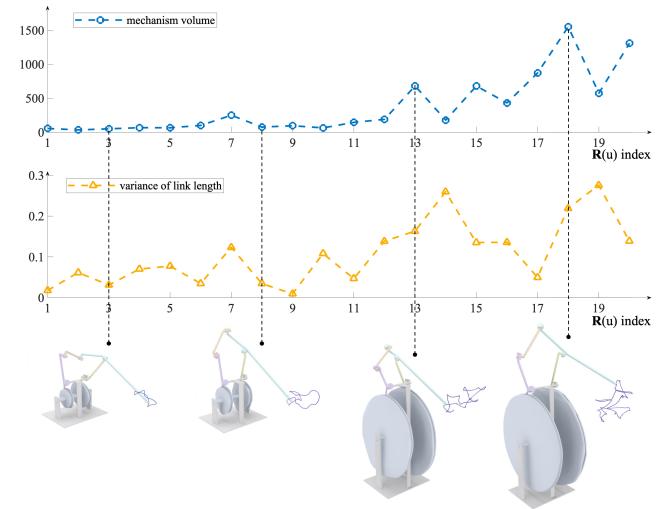


Fig. 11. The mechanism volume (top) and variance of normalized link length (middle) for generating 20 3D paths with increasing shape complexity, as well as four example input curves and corresponding mechanisms (bottom).

constraint between the two 3D cams (i.e., Equation 19), we require a sufficient distance between the two active link joints J_1 and J_5 along the x -axis when choosing the linkage's geometry; see again Figure 4. Figure 10(c) shows the resulting 3D cam-linkage mechanism for generating the input curve with arbitrary shape and 11 C^0 points. We further optimize the 3D cam-linkage mechanism's geometry to reduce its weight and smooth its motion (i.e., Equation 12); see Figure 10(d). Please refer to the supplementary material for more details about this experiment.

Generating 3D paths with varying shape complexity. We conduct a quantitative experiment to evaluate the impact of the input curve's shape complexity on our designed 3D cam-linkage mechanism. In detail, we model 20 continuous, closed curves $\{R_k\}$, $1 \leq k \leq 20$ with increasing shape complexity using the particle-based method mentioned above, where $\text{Length}(R_{k+1}) = 1.1 \cdot \text{Length}(R_k)$ and R_k has $k - 1 C^0$ points. Taking each curve R_k as an input, we design a 3D cam-linkage mechanism using the optimization method in Section 5. Figure 11 shows the total volume (excluding the support structure) and the variance of normalized link length of each mechanism when the shape complexity of the input curve is increasing. We can see that generating a curve with more complex shape requires a mechanism with a larger volume since two larger 3D cams (with a longer pitch curve) are necessary to encode the input curve with more complex shape. In addition, generating a more complex input curve also results in a larger variance of normalized link length. Please refer to the supplementary material for all the 20 input curves and corresponding mechanisms.

Fabrication. To evaluate the kinematic performance of our mechanism, we used our approach to design a 3D cam-linkage mechanism that can generate a TENNIS BALL SEAM curve and fabricated it with 3D printing; see Figure 12. In detail, we 3D printed the 3D cams and 5-bar linkage as a non-assembly, articulated model (i.e., a single

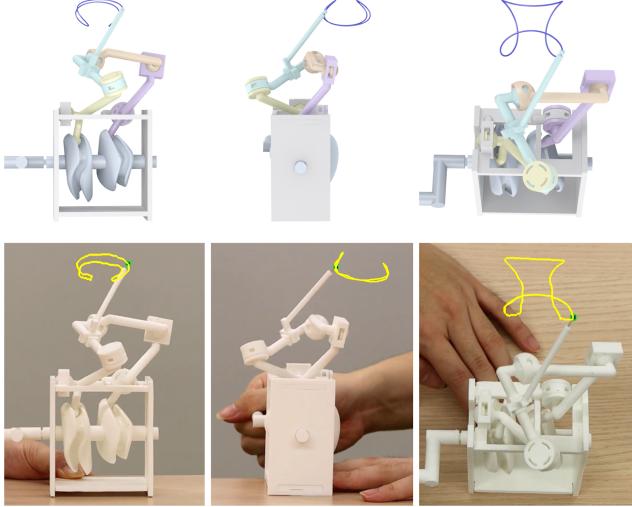


Fig. 12. Evaluating the kinematic performance of our mechanism. (Top) Our designed mechanism in front, side, and top views, respectively. (Bottom) Corresponding views of our 3D printed prototype, where the generated path is tracked on video images and visualized in yellow.

piece) using the Ultimaker S3 printer with Tough PLA material and water-soluble PVA support material, where the link joint tolerance is set as 0.5 mm and the cam joint tolerance is set as 0.3 mm. The support structure is partitioned, 3D printed as multiple parts, and then assembled with the single articulated piece of 3D cams and 5-bar linkage. We tracked the path generated by our 3D printed mechanism using an image-based approach. Figure 12 compares our design and the 3D printed prototype in front, side, and top views, respectively. We can see that the 3D print matches the virtual design very well, and the tracked curve is very close to the input curve in all the three views. The discrepancy between the two curves can be caused by several reasons including fabrication tolerance, material deformation, and tracking inaccuracy, which can be reduced by using a more precise manufacturing technique, tougher fabrication material (e.g., steel), and a better curve tracking technique.

To verify that our 3D cam-linkage mechanism is able to generate a path with C^0 points, we designed a mechanism that can draw a

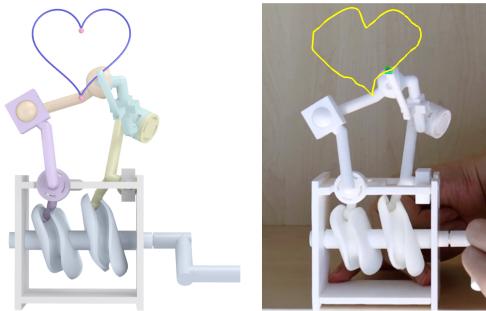


Fig. 13. Evaluating the kinematic performance of our mechanism for generating a planar HEART curve with two C^0 points.

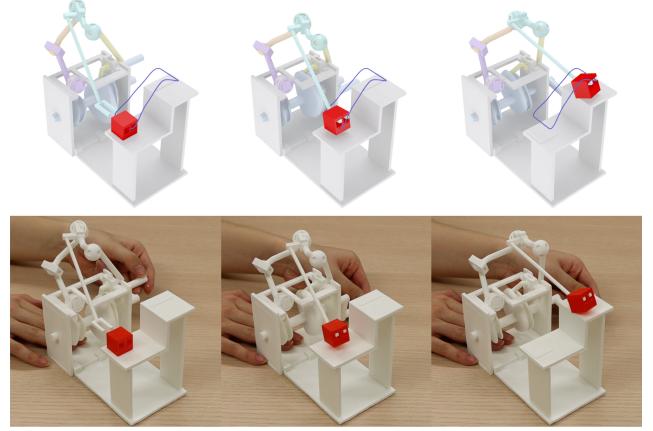


Fig. 14. Designing and fabricating a low-cost manipulator to perform a pick-and-place task by using our 3D cam-linkage mechanism.

planar HEART curve with two C^0 points, 3D printed it using the same 3D printer and settings as above, and tracked the generated path using the image-based approach. Figure 13 shows that the curve as well as its two C^0 points generated by the 3D printed mechanism are very close to the virtual counterpart. Moreover, the 3D printed mechanism's end-effector stops when passing each of the two C^0 points even though the 3D cams rotate uniformly, which is consistent with our kinematic simulation of the mechanism; see the accompanying video for a demo. The reason that our 3D cam-linkage mechanism is able to reproduce C^0 points in the prescribed 3D curve is that the two 3D cams (with a smooth groove) are able to control the motion of the two driving links (i.e., followers) of the 5-bar linkage respectively such that both driving links have zero speed when the end-effector is passing each C^0 point; see again Equation 15.

We fabricated one more prototype to demonstrate that our mechanism can be used as a low-cost manipulator driven by a single actuator to perform pick-and-place tasks repetitively; see Figure 14. In this experiment, the task is to pick a box from the floor, manipulate the box, place it on a table, go back to pick another box, and so on and so forth. The path is designed as a closed 3D spline curve with two approximately straight segments for the mechanism's end-effector to pick and place the box, respectively. To avoid collision between the end-effector and the box, we add one more energy term in our optimization to approximately control the end-effector's orientation for 20 sampled points on the path segments when picking and placing the box. Figure 14 compares three states between the real manipulation process and the virtual counterpart, and we can see that they all match each other very well.

Generalizing 3D cam-linkage for 3D motion generation. Our 3D cam-linkage mechanism can be generalized for exact 3D motion generation, which is a more complex motion transfer task than exact 3D path generation. To this end, we first design a 6-DOF 8-bar spatial linkage and then reduce the linkage's DOFs from 6 to 1 by composing the linkage with three 3D cam-follower mechanisms. We extend our approach in Section 5 to design the generalized mechanism,

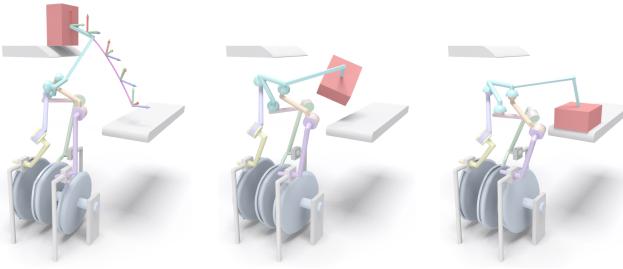


Fig. 15. Generalizing our 3D cam-linkage mechanism for exact 3D motion generation, which consists of three 3D cams and eight links.

where the user input is a continuous set of rigid transformations represented as a continuous 3D path as well as a local frame for each point on the path; see Figure 15 (left). We solve the design problem by making two modifications on Algorithm 1. First, we fix instead of searching for the topology of the generalized mechanism with eight links and three cams. Second, we modify the IK solver to compute the link joint angles of the 6-DOF 8-bar linkage according to the prescribed positions and orientations of the end-effector. Figure 15 shows an example mechanism for 3D object manipulation. Different from the manipulator in Figure 14, the generalized mechanism can exactly control both the position and orientation of the object along the whole path.

7 CONCLUSION

Exact 3D path generation is a fundamental yet challenging problem in many fields such as mechanical engineering, manufacturing, and robotics. To address this problem, we propose and study a new 3D cam-linkage mechanism that consists of only seven mechanical parts in this paper. A family of computational techniques has been developed to model and design the 3D cam-linkage mechanism, where the key idea is to consider the 3D cam-linkage mechanism as a composition of two 3D cam-follower mechanisms and a 3-DOF 5-bar linkage. Our 3D cam-linkage mechanism has been evaluated in many aspects including its topology, kinematics, ability to generate 3D paths with varying shape complexity, and a potential application for 3D object manipulation.

Limitations and Future Work. Our work has several limitations that open up interesting directions for future work. First, we do not model the dynamics and friction of our 3D cam-linkage mechanism, which will be useful for understanding how much workload the end-effector can carry in practice, e.g., in the manipulation task. Second, we do not study the role of backlash in our 3D cam-linkage mechanisms; for example, even though our designed mechanism stays away from singular configurations, the physical mechanism could still get stuck due to backlash. Third, our 3D cam-linkage mechanism currently has a fixed topology with two 3D cams and five links. We consider developing a general algorithm to model and design 3D cam-linkage mechanisms with m cams (e.g., $m \in \{2, 3\}$) and n links (e.g., $n \in \{5, 6, 7, 8\}$) for 3D path/motion generation as an interesting future work. Lastly, our mechanism currently only has a single end-effector, limiting the tasks that it can accomplish. To address this limitation, one possible way is to combine multiple

3D cam-linkage mechanisms, possibly with other mechanical parts such as non-circular gears [Xu et al. 2020], into a more powerful yet more complex mechanism.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments, Ke Chen for proofreading the paper, and Robert Ferréol for providing equations of some input parametric curves. This work was supported by the SUTD Start-up Research Grant (Number: SRG ISTD 2019 148), and the National Natural Science Foundation of China (62025207).

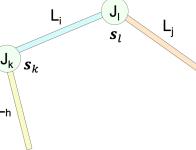
REFERENCES

- Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: Interactive Linkage Editing using Symbolic Kinematics. *ACM Trans. on Graph. (SIGGRAPH)* 34, 4 (2015), 99:1–99:8.
- Shaoping Bai, Zhongyi Li, and Jorge Angeles. 2022. Exact Path Synthesis of RCCC Linkages for a Maximum of Nine Prescribed Positions. *Journal of Mechanisms and Robotics* 14, 2 (2022), 021011:1–021011:8.
- Gaurav Bharaj, Stelian Coros, Bernhard Thomaszewski, James Tompkin, Bernd Bickel, and Hanspeter Pfister. 2015. Computational Design of Walking Automata. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 93–100.
- Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and Fabricating Mechanical Automata from Mocap Sequences. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 186:1–186:11.
- Yingjie Cheng, Yucheng Sun, Peng Song, and Ligang Liu. 2021. Spatial-Temporal Motion Control via Composite Cam-follower Mechanisms. *ACM Trans. on Graph. (SIGGRAPH Asia)* 40, 6 (2021), 270:1–270:15.
- Jinkui Chu and Jianwei Sun. 2010. A New Approach to Dimension Synthesis of Spatial Four-Bar Linkage Through Numerical Atlas Method. *Journal of Mechanisms and Robotics* 2, 4 (2010), 041004:1–041004:14.
- Wen-Yean Chung. 2015. Synthesis of Spatial Mechanism UR-2SS for Path Generation. *Journal of Mechanisms and Robotics* 7, 4 (2015), 041009:1–041009:9.
- Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 83:1–83:12.
- Clement Gosselin and Jorge Angeles. 1990. Singularity Analysis of Closed-loop Kinematic Chains. *IEEE Transactions on Robotics and Automation* 6, 3 (1990), 281–290.
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A Simple C++ Geometry Processing Library. <https://libigl.github.io/>.
- F. J. Kay and R. E. Haws. 1975. Adjustable Mechanisms for Exact Path Generation. *Journal of Engineering for Industry* 97, 2 (1975), 702–707.
- James Kennedy and Russell Eberhart. 1995. Particle Swarm Optimization. In *Proc. Int. Conf. on Neural Networks*. 1942–1948.
- Z. Li, G. Nawratil, F. Rist, and M. Hensel. 2020. Invertible Paradox Loop Structures for Transformable Design. *Comp. Graph. Forum (Eurographics)* 39, 2 (2020), 261–275.
- Wenrui Liu, Jianwei Sun, and Jinkui Chu. 2020. Synthesis of a Spatial RRSS Mechanism for Path Generation Using the Numerical Atlas Method. *Journal of Mechanical Design* 142, 1 (2020), 012303:1–012303:12.
- Dominik Mannhart, Fabio Dubois, Karen Bodie, Victor Klemm, Alessandro Morra, and Marco Hutter. 2020. CAMI - Analysis, Design and Realization of a Force-Compliant Variable Cam System. In *Proc. IEEE Int. Conf. on Robotics and Automation*. 850–856.
- Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. 2014. ChaCra: An Interactive Design System for Rapid Character Crafting. In *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. 123–130.
- J. P. Merlet. 2006. Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots. *Journal of Mechanical Design* 128, 1 (2006), 199–206.
- D. Mundo, G. Gatti, and D.B. Dooner. 2009. Optimized Five-bar Linkages with Non-circular Gears for Exact Path Generation. *Mechanism and Machine Theory* 44, 4 (2009), 751–760.
- D. Mundo, J. Y. Liu, and H. S. Yan. 2006. Optimal Synthesis of Cam-Linkage Mechanisms for Precise Path Generation. *Journal of Mechanical Design* 128, 6 (2006), 1253–1260.
- G. Nishida, A. Bousseau, and D. G. Aliaga. 2019. Multi-Pose Interactive Linkage Design. *Comp. Graph. Forum (Eurographics)* 38, 2 (2019), 277–289.
- Zherong Pan, Min Liu, Xifeng Gao, and Dinesh Manocha. 2019. Globally Optimal Joint Search of Topology and Trajectory for Planar Linkages. In *International Symposium on Robotics Research*. 1–16.
- Robin Roussel, Marie-Paule Cani, Jean-Claude Léon, and Niloy J. Mitra. 2018. Exploratory Design of Mechanical Devices with Motion Constraints. *Comp. & Graph. (Computational Fabrication)* 74 (2018), 244–256.

- Kevin Russell, Qiong Shen, and Raj S. Sodhi. 2014. *Mechanism Design: Visual and Programmable Approaches*. CRC Press.
- Yesh Pal Singh. 1981. Synthesis of Cam-Link Mechanisms for Exact Path Generation. *Mechanism and Machine Theory* 16, 4 (1981), 447–457.
- Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J. Mitra. 2017. Computational Design of Wind-up Toys. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 238:1–238:13.
- J. W. Sun, D. Q. Mu, and J. K. Chu. 2012. Fourier Series Method for Path Generation of RCCC Mechanism. *Proc. of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 226, 3 (2012), 816–827.
- Takuto Takahashi and Hiroshi G. Okuno. 2018. Design and Implementation of Programmable Drawing Automata based on Cam Mechanisms for Representing Spatial Trajectory. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 450–455.
- Bernhard Thomaszewski, Stelian Coros, Damien Gouge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-Based Characters. *ACM Trans. on Graph. (SIGGRAPH)* 33, 4 (2014), 64:1–64:9.
- John J. Uicker, Gordon R. Pennock, and Joseph E. Shigley. 2016. *Theory of Machines and Mechanisms (5th Edition)*. Oxford University Press.
- C. W. Wampler, A. P. Morgan, and A. J. Sommese. 1992. Complete Solution of the Nine-Point Path Synthesis Problem for Four-Bar Linkages. *Journal of Mechanical Design* 114, 1 (1992), 153–159.
- Hao Xu, Tianwen Fu, Peng Song, Mingjun Zhou, Chi-Wing Fu, and Niloy J. Mitra. 2020. Computational Design and Optimization of Non-Circular Gears. *Comp. Graph. Forum (Eurographics)* 39, 2 (2020), 399–409.
- Hongyi Xu, Espen Knop, Stelian Coros, and Moritz Bächer. 2018. Bend-It: Design and Fabrication of Kinetic Wire Characters. *ACM Trans. on Graph. (SIGGRAPH Asia)* 37, 6 (2018), 239:1–239:15.
- Tsuneo Yoshikawa. 1985. Dynamic Manipulability of Robot Manipulators. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1033–1038.
- Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, and Baining Guo. 2012. Motion-Guided Mechanical Toy Modeling. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 127:1–127:10.

APPENDIX

We describe how we formulate the constraint C_L in Equation 4 for our 5-bar spatial linkage. For each link L_i , $1 \leq i \leq 5$, we denote its two neighboring links as L_h and L_j , among which L_i and L_h are connected by joint J_k while L_i and L_j are connected by joint J_l ; see the inset. At time t , the pose of each link L_i can be calculated based on its neighboring link's pose and the corresponding joint angle. For example, we calculate the link L_i 's pose $\mathbf{M}_{L_i}(t)$ based on its neighboring link L_h 's pose $\mathbf{M}_{L_h}(t)$ and the joint angle $s_k(t)$:



$$\mathbf{M}_{L_i}(t) = \mathbf{M}_{J_k}^{L_i} \mathbf{M}(s_k(t)) \mathbf{M}_{L_h}^{J_k} \mathbf{M}_{L_h}(t) \quad (23)$$

where $\mathbf{M}(s_k(t))$ is the matrix representation of the joint angle $s_k(t)$, $\mathbf{M}_{L_h}(t)$ represents the link L_h 's pose with respect to its center, and $\mathbf{M}_{L_h}^{J_k}$ is a constant matrix that represents a translation from the joint J_k 's center to the link L_h 's center. To simplify the notation, we re-write Equation 23 as:

$$\mathbf{M}_{L_i}(t) = \tilde{\mathbf{M}}(s_k(t)) \mathbf{M}_{L_h}(t) \quad (24)$$

where $\tilde{\mathbf{M}}(s_k(t)) = \mathbf{M}_{J_k}^{L_i} \mathbf{M}(s_k(t)) \mathbf{M}_{L_h}^{J_k}$.

Since our 5-bar spatial linkage is a single loop of five joints, we concatenate Equation 24 for every joint in the loop and get:

$$\tilde{\mathbf{M}}(s_5(t)) \tilde{\mathbf{M}}(s_4(t)) \tilde{\mathbf{M}}(s_3(t)) \tilde{\mathbf{M}}(s_2(t)) \tilde{\mathbf{M}}(s_1(t)) = \mathbf{I} \quad (25)$$

Hence, the constraint denoted as \mathbf{M}_L can be written as:

$$\mathbf{M}_L = \tilde{\mathbf{M}}(s_5(t)) \tilde{\mathbf{M}}(s_4(t)) \tilde{\mathbf{M}}(s_3(t)) \tilde{\mathbf{M}}(s_2(t)) \tilde{\mathbf{M}}(s_1(t)) - \mathbf{I} = \mathbf{0} \quad (26)$$

We obtain the constraint C_L by concatenating the 4×4 matrix \mathbf{M}_L (ignoring the last row) into a 12×1 column vector.

To calculate the matrix $\mathbf{M}_{L_2}(t)$ in Equation 6, we concatenate Equation 24 for links 1 and 2, starting from the fixed link 5:

$$\mathbf{M}_{L_2}(t) = \tilde{\mathbf{M}}(s_2(t)) \tilde{\mathbf{M}}(s_1(t)) \mathbf{M}_{L_5} \quad (27)$$

where \mathbf{M}_{L_5} represents the pose of the fixed link L_5 .