EUROGRAPHICS 2019 / P. Alliez and F. Pellacini
(Guest Editors)

Volume 38 (2019), Number 2

Multi-Pose Interactive Linkage Design

Teaser

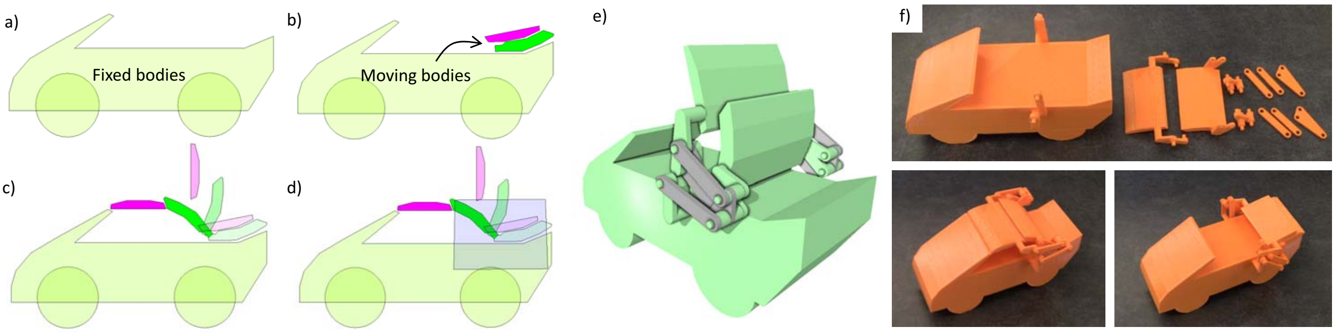

Figure 1: ***Interactive design of a multi-pose and multi-body linkage.*** *Our system relies on a simple 2D drawing interface to let the user design a multi-body object including (a) fixed bodies, (b) moving bodies (green and purple shapes), (c) multiple poses for each moving body to specify the desired motion, and (d) a desired region for the linkage mechanism. Then, our system automatically generates a 2.5D linkage mechanism that makes the moving bodies traverse all input poses in a desired order without any collision (e). The system also automatically generates linkage parts ready for 3D printing and assembly (f). Please refer to the accompanying video for a demonstration of the sketching interface and animations of the resulting mechanisms.*

Abstract

We introduce an interactive tool for novice users to design mechanical objects made of 2.5D linkages. Users simply draw the shape of the object and a few key poses of its multiple moving parts. Our approach automatically generates a one-degree-of-freedom linkage that connects the fixed and moving parts, such that the moving parts traverse all input poses in order without any collision with the fixed and other moving parts. In addition, our approach avoids common linkage defects and favors compact linkages and smooth motion trajectories. Finally, our system automatically generates the 3D geometry of the object and its links, allowing the rapid creation of a physical mockup of the designed object.

CCS Concepts
• ***Computing methodologies*** → ***Shape modeling;***

1. Introduction

With the proliferation of rapid-prototyping technology, anyone can now engage in the creation of personalized mechanical objects, far beyond what is achievable with popular construction kits like *Lego Technic*. Yet, designing a mechanical object that performs a desired motion raises multiple challenges for novices. First, small changes in the mechanism geometry can result in large changes in the resulting motion, making design exploration a tedious, counter-intuitive task. Second, many linkage setups for multi-pose and multi-body objects encounter some forms of defects or collisions during their animation. We propose an interactive system that hides such challenges from the user, who can then focus on designing both the shape and the motion of an object composed of multiple moving bodies connected by automatically-computed linkages.

We created our system to provide several important features for interactive design:

Shape and motion control. We need a user interface that allows the rapid specification of 3D shapes and their motion, even for novices with limited knowledge in 3D modeling and animation. We achieve this goal thanks to a simple sketch-based interface, where users draw the 2D profile of each moving part, as well as its position and orientation in a few key poses (Figure 1a-d). These 2D shapes are then extruded in depth to obtain a 2.5D multi-body object. Our system supports a varying number of poses, so that users can iteratively refine complex trajectories by inserting new poses. In addition, we also offer a global parameter to control the smoothness of the trajectories and balance it against the accuracy with which the moving bodies traverse the specified poses.

Mechanism size and location control. Given the sketched input, our approach automatically connects the moving and fixed bodies with a generated linkage that makes the moving bodies traverse their poses in the specified order without colliding with the fixed and other moving bodies (Figure 1e). We provide two user controls on this linkage synthesis. First, we allow users to indicate a region where the linkage should be placed to satisfy aesthetic or usability goals. Second, we offer a global parameter to control the size of the mechanism, which again competes with the smoothness and accuracy of the resulting trajectories.

Rapid prototyping. Finally, we seek to allow rapid testing of the generated linkage in the physical world. We achieve this goal by designing parametric linkage geometry that is ready for 3D printing and assembly without additional fixtures. In addition, our system also avoids various common linkage defects to guarantee that our result is functional. Our current implementation supports multi-bar linkages (e.g., 4-bar, 6-bar, 8-bar linkages, etc.) as well as the popular slider crank.

To satisfy the above requirements, we have created a novel multi-bar linkage method that includes an optimization with multiple concurrent goals. Soft terms measure the accuracy of the moving body trajectory with respect to the specified poses, the smoothness of this trajectory, the compactness of the mechanism, and the location of the mechanism within the user-specified region. Hard constraints ensure that the linkage is free of common defects, such as wrong ordering of the traversed poses, alignment of consecutive links during motion, disconnected pose space, as well as any collision between the moving and fixed bodies or between the links. Unfortunately, several of these hard constraints require running a kinematic simulation of the linkage with actual 3D geometry, and as such cannot be integrated in standard gradient-based solvers. Instead, we adopt a stochastic exploration of the solution space, where we first generate a large number of linkages that approximately satisfy the specified poses, then reject the ones that violate the hard constraints, and finally select the valid linkage that best satisfies all soft terms. We accelerate this exploration using a coarse-to-fine refinement methodology based on particle filter optimization [LCS16], thus providing users with valid designs after only a few seconds of computation.

In summary, our contributions include:

- *Interactive Design.* An end-to-end approach to create physical 2.5D objects made of multiple moving bodies connected by linkages. Our system targets novice users and lets them design dynamic objects simply by sketching both the shape and the pose of the moving parts. We offer fine control on the trajectory of the moving bodies by supporting multiple poses, and global control on the trade-offs between compactness of the mechanism, smoothness of the trajectories, and accuracy with respect to the input poses.
- *Practical and Defect-Free Solution.* A stochastic optimization that efficiently finds a well-behaved linkage that satisfies the user goal while avoiding defects and collisions. While not real-time, this optimization is fast enough to let users iterate on a design within seconds.
- *Automatic Synthesis.* An automatic linkage synthesis procedure that outputs geometry of the mechanism ready for fabrication and assembly.

2. Background and Related Work

The goal of our interactive approach is to help casual users create fabrication-ready 2.5D multi-body mechanical objects composed of planar linkages. We first provide a brief introduction to planar linkages, and then discuss related work on linkage synthesis from the engineering literature and on interactive mechanical design from the computer graphics literature.

Planar linkages. In our context, a linkage is a mechanism composed of rigid links, also called *bars*, connected by joints. Furthermore, we focus on two types of one-degree-of-freedom joints – the rotary hinge, also called *revolute joint*, and the linear slider, also called *prismatic joint*. Linkage mechanisms are commonly classified according to the number of links they are composed of and the types of their joints, which are denoted with the letter R for revolute and P for prismatic. Our system supports multi-bar linkages with K moving bodies that are connected by revolute joints (Figure 3). For example, for $K = 1$ this mechanism is reduced to the 4R four-bar linkage, and for $K = 2$ it becomes the Watt-I type six-bar linkage, which appear in many everyday objects [HD64, SE84, MS11]. Our system also supports the RRRP linkage or *slider crank*, which is a rotating link connected to a translating slider using three revolute joints and one prismatic joint. We refer the interested reader to the book by McCarthy and Soh [MS11] for additional details on different types of linkages. In what follows, we refer to the moving parts of an object as the *moving bodies*, which are connected to a *fixed body* by the linkage.

Linkage synthesis. The seminal Burmester theory [Bur86] is a graphical method to solve for the parameters of a planar 4R four-bar linkage for up to five input poses of the moving body. Later work expressed the geometric constraints induced by the lengths of the links as a system of equations [SR67, San59]. However, such methods disregard other desired properties of a linkage, such as respecting the order of visited poses (order defect) or avoiding singular configurations that occur when two adjacent bars are collinear (branch defect) or when different poses cannot be reached without disassembly (circuit defect), as reviewed by Balli and Chand [BC02]. Multiple extensions to Burmester's theory have been proposed to address some of these defects [Fil72, Wal76, CF91, CM93, MC95, PM14, PMW14]. Unfortunately, most solutions are specific to a particular type of defect and a particular type of linkage. More importantly, these geometric methods assume that the designer provides exact input poses, and produce solutions that respect these poses with high precision [CKE13]. In contrast, we target a casual design scenario where the input poses are approximately specified by rapid sketching. Our system thus seeks to balance the precision of the pose traversal with other objectives such as the smoothness of the trajectory and compactness of the linkage.

Our need to satisfy multiple concurrent objectives makes our work most similar to optimization-based linkage synthesis methods. Various optimization strategies have been considered, such as evolutionary algorithms [BD09], simulated annealing [MA08], non-convex optimization [GLC*16], and differential evolution [GM12, KN17]. However, many of these approaches target the *path generation* problem, where the goal is to generate a linkage whose end-effector follows a given trajectory. We target the different problem of *motion generation* or *body guidance* [MS11], where the goal is to make a moving body traverse a given set of poses, each defined by a position and an orientation. In addition, most methods ignore the linkage defects mentioned above, as well as collisions. One exception is the work by Zheng et al. [ZSC16], who propose a collision detection algorithm for a fold-able scissor linkage system. However, their approximate solution is specific to collisions between the thin bars of scissor linkages, while we need to detect collisions between a more general class of objects including arbitrary rigid bodies and a wider family of linkages. We address these challenges with an interactive design system that uses a custom stochastic optimization having soft energy terms to measure the quality of a linkage and hard constraints to prevent any defect or collision.

Interactive mechanical design. The computer graphics community has proposed multiple interactive mechanical design systems. Motivated by applications in character animation, many methods take an articulated model as input and focus on reproducing a pre-defined animation [ZXS*12, TCG*14, CLM*13]. In contrast, our *body guidance* approach lets users sketch disconnected fixed and moving bodies and generates a linkage that connects them such that the moving bodies traverse a set of desired poses.

Our method is most related to the system by Megaro et al. [MTG*14], which allows users to create linkages by specifying the two extreme poses of a set of moving bodies. While we take inspiration from their design workflow, our goal to support multiple poses of 2.5D bodies requires a different solution addressing new challenges. First, the method needs to consider a much larger set of candidate linkages to produce the complex trajectories dictated by multiple poses, which can include abrupt changes of orientation or direction of the moving bodies. Second, Megaro et al. focus on planar objects, which allows them to resolve collisions by offsetting mechanical components in depth. In contrast, we target 2.5D objects with co-planar moving bodies, which requires us to run kinematic simulations during optimization for collision avoidance. Third, while the algorithm by Megaro et al. automatically generates a single solution, our approach incorporates soft objectives, such as the smoothness of the trajectory and the compactness of the mechanism, which provides users a larger design space. In addition, these soft objectives act as regularization terms against the specified poses, which users only need to specify approximately. Finally, while Megaro et al. describe how to prevent alignment of consecutive links on part of their mechanisms, their algorithm to generate connectors between the moving bodies can suffer from defects that our method avoids, as detailed in our evaluation. Our method also needs to account for order defect, which does not occur with only two poses.

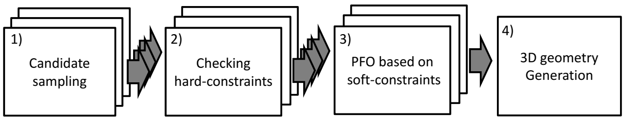

Figure 2: ***System pipeline.*** *Our approach starts with 1) a random sampling of mechanisms based on $E_{rigid}(\boldsymbol{x})$. 2) For each candidate, the hard-constraints, $C_{defect}(\boldsymbol{x})$ and $C_{coll}(\boldsymbol{x})$, are checked. Candidates that do not satisfy the hard-constraints are discarded. 3) A particle filter optimization is performed based on soft-constraints, $E_{acc}(\boldsymbol{x})$, $E_{smooth}(\boldsymbol{x})$, $E_{loc}(\boldsymbol{x})$, and $E_{size}(\boldsymbol{x})$. 4) Finally, the best candidate is selected, and its fabrication-ready 3D geometry is generated.*

Our work also relates to the system by Coros et al. [CTN*13], where users sketch the motion curves of various actuation points on an articulated character provided as input. Their work is an instance of the *path generation* problem. While the position and orientation of a moving body could in theory be specified by two point trajectories, path generation methods do not generalize to the problem of body guidance because these two trajectories should be perfectly synchronized to maintain the rigidity and orientation of the moving body – this is almost impossible to do manually.

While our goal is to generate a linkage from scratch, several methods have been proposed to modify existing linkages. In particular, Bächer et al. [BCT15] allows users to edit a working linkage while preserving its functionality, while Megaro et al. [MZB*17] convert rigid linkages into compliant mechanisms made of soft flexures.

Finally, most of the above methods focus on the design of articulated characters, while we target the design of everyday objects composed of multiple moving parts. We thus share the goal of Koo et al. [KLY*14], who propose a modeling system to help designers create functional prototypes of mechanical objects. However, our system supports more complex motions thanks to multi-bar linkages. In addition, the system by Koo et al. can generate mechanisms that have more than one degree of freedom, which requires the user to move multiple links synchronously to achieve a target motion. In contrast, our approach produces mechanisms that have only one degree of freedom, so it is easy for users to animate the fabricated object while maintaining the desired orientation.

3. Overview

The input to our method is a set of vector drawings representing fixed and moving parts of a rigid object, in several desired poses. Please refer to our accompanying video for several example design sessions using our sketching interface. Given this input, our goal is to generate rigid, one-degree-of-freedom mechanisms that enable the moving bodies to traverse all poses in a desired order. Out of the many solutions that can perform this task, we favor the ones that are compact and that yield a smooth trajectory. We express these design goals with four soft constraints represented as energy

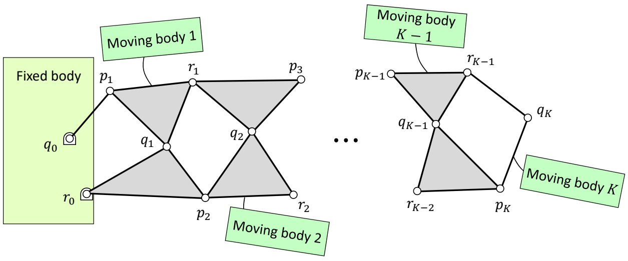

Figure 3: ***Multi-bar linkage.*** *Our mechanisms consist of a fixed body and K moving bodies connected by a linkage assembly that has a single degree of freedom. Gray triangles represent rigid ternary links, each of which has three revolute joints. The mechanism consists of K four-bar loops, q_{k-1}, r_{k-1}, q_k, and p_k, where the index k varies from 1 to K for p_k, from 0 to K for q_k, and from 0 to $K-1$ for r_k. For $K=1$, the mechanism reduces to a 4R four-bar linkage, and for $K=2$, the mechanism becomes a Watt-I six-bar linkage [MS11].*

terms: i) the accuracy of the trajectory with regard to the input poses E_{acc}, ii) the smoothness of the trajectory E_{smooth}, iii) the distance of the mechanisms from a user-specified region at the first pose E_{loc}, and iv) the size of the mechanism E_{size}. In addition, we define two hard constraints: i) C_{coll} – a valid mechanism should not produce any collision between the rigid bodies, nor between the links and joints, and linkage should also not collide with any safety region that users optionally indicate, and ii) C_{defect} – the linkage should not have any defect. Combining these terms gives the following constrained optimization

$$\begin{aligned} \arg\min_{\mathbf{x}} \; & w_{acc}E_{acc}(\mathbf{x}) + w_{smooth}E_{smooth}(\mathbf{x}) \\ & + w_{loc}E_{loc}(\mathbf{x}) + w_{size}E_{size}(\mathbf{x}) \\ \text{subject to} & \begin{cases} C_{coll}(\mathbf{x}) \\ C_{defect}(\mathbf{x}) \end{cases} , \end{aligned} \tag{1}$$

where $\mathbf{x}$ denotes the set of parameters of the linkage mechanism (see Section 4.2 for more details).

However, the soft and hard constraints of Equation 1 are highly nonlinear and also include terms that require a kinematic simulation to be evaluated (i.e., for $E_{smooth}(\mathbf{x})$ and $C_{coll}(\mathbf{x})$). We tackle this challenge by decomposing the problem into multiple steps, as illustrated in Figure 2. First, we generate many candidate solutions by mildly perturbing the poses of the sketched bodies and for each computing a multi-bar linkage that passes through the perturbed poses (Section 4.1). This allows the system to consider solutions that might increase E_{acc} but decrease the other soft constraints. In a second step, the candidates are checked against the hard constraints (Section 5). Our approach considers order, branch, and circuit defects as well as collisions. In a third step, a particle filter optimization is used to refine candidates, and the overall best solution in terms of the soft constraints is selected (Section 6). Finally, we generate 3D models of the linkage as well as the fixed and moving bodies that are ready to fabricate (Section 7).

4. Candidate Sampling

The first step of our method is to generate a large number of candidate rigid mechanisms that approximately satisfy the input poses. Figure 3 illustrates our general multi-bar mechanism, where each moving body is attached to a fixed-length link, and is connected to adjacent bodies by links and revolute joints that form four-bar loops. For example, the left most four bars, q_0, p_1, q_1, and r_0, form a four-bar loop where the first moving body is attached to the link p_1 r_1. Similarly, the k-th moving body is attached to the k-th four-bar loop, q_{k-1}, r_{k-1}, q_k, and p_k, and so forth. For K moving bodies, our linkage mechanism has $3K+1$ joints that consist of $p_k(k=1,\cdots,K)$, $q_k(k=0,\cdots,K)$, and $r_k(k=0,\cdots,K-1)$. Since the linkage mechanism can be uniquely defined by the initial xy coordinates of the joints, there are $6K+2$ variables to determine.

Megaro et al. [MTG*14] proposed to generate linkages by sampling random joints along the boundary of the rigid bodies and by selecting pairs of links between adjacent bodies such that the links have minimal length variation between poses (two in their case). However, we found that this strategy often results in linkages that perform poorly with regards to our soft and hard constraints (see Section 8.6 for a comparison to Megaro et al. [MTG*14]). We next describe a more general sampling strategy that accounts for multiple, approximate poses, and that considers joints away from the boundary of the rigid bodies.

4.1. Sampling

Our sampling approach is to augment the solution space by perturbing the input poses, such that candidate solutions admit deviation from the input poses, which can be balanced later against the other soft and hard constraints. Since the first and last poses are typically more important than the intermediate poses, we only perturb the intermediate poses by adding a Gaussian noise with an empirically-determined standard deviation $\sigma = 0.5d$, where d is the shortest length of the bounding box of each moving body. We analyze the impact of different values of σ in Section 8.4, but use $\sigma = 0.5d$ for all results in this paper. We also make sure that the perturbed moving bodies do not collide with other moving and fixed bodies. Figure 4 illustrates our sampling process for $K=1$ moving bodies (i.e., a four-bar linkage).

4.2. Multi-bar linkage

For each perturbation, our method seeks to generate a rigid multi-bar linkage system that best traverses the N perturbed poses of the K moving bodies $T_k^{i\prime}(i=1,\cdots,N, k=1,\cdots,K)$, which are represented by their transformation matrices relative to the fixed body:

$$T_k^{i\prime} = \begin{bmatrix} \cos\theta_k^{i\prime} & -\sin\theta_k^{i\prime} & u_k^{i\prime} \\ \sin\theta_k^{i\prime} & \cos\theta_k^{i\prime} & v_k^{i\prime} \\ 0 & 0 & 1 \end{bmatrix} . \tag{2}$$

We initialize the solution by randomly sampling all the joints at the first pose, $p^1_{1\cdots K}, q^1_{0\cdots K}, r^1_{0\cdots K-1}$. The coordinates of the joints at other poses are calculated as $p_k^i = T_k^{i\prime}(T_k^{1\prime})^{-1}p_k^1$, $q_k^i = T_k^{i\prime}(T_k^{1\prime})^{-1}q_k^1$, and $r_k^i = T_k^{i\prime}(T_k^{1\prime})^{-1}r_k^1$, respectively. Since the matrices $T_k^{i\prime}$ express transformations relative to the fixed body, the

Sampling for four-bar linkage

Four-bar linkage & slider crank

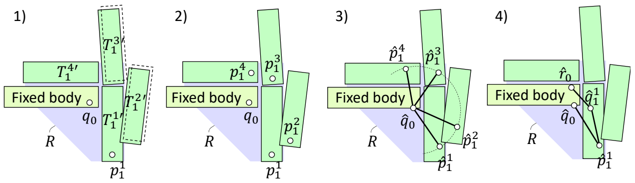

Figure 4: ***Illustration of our sampling mechanism with*** $K =$ 1 ***moving bodies.*** *1) The intermediate $N-2$ poses (dotted rectangles are perturbed to $T_1^{i\prime}$, and the position of the moving joint at the firs pose, p_1^1, and its corresponding fixed point q_0 are sampled within a user-specified desired region R (blue region). 2) The positions of moving point at other poses are calculated as $p_1^i = T_1^{i\prime}(T_1^{1\prime})^{-1}$ 3) The moving point p_1^1 and the fixed point q_0 are optimized to and $\hat{q}_0$ by minimizing the deviation of the link length between moving point and the fixed point. The resulting link $\hat{q}_0\hat{p}_1^1$ constitu a driving link of the four-bar linkage at the first pose. 4) Anot moving point at the first pose q_1^1 and its corresponding fixed po r_0 are sampled in a similar manner for a follower. These two li form a four-bar linkage.*

transformation matrix for the fixed body $T_0^{i\prime}$ is the identity mat However, the length of the links obtained with this random sa pling is not necessarily constant across different poses. Therefo we optimize the initial joints p_k^1, q_k^1, and r_k^1 by minimizing

$$E_{rigid}(p^1_{1\cdots K}, q^1_{0\cdots K}, r^1_{0\cdots K-1}) = \sum_{i=2}^{N}\Big[\Delta l(p_1^i, q_0^i) + \Delta l(q_1^i, r_0^i)$$

$$+ \sum_{k=2}^{K}\Big(\Delta l(p_k^i, r_{k-2}^i) + \Delta l(p_k^i, q_{k-1}^i) + \Delta l(q_k^i, r_{k-1}^i)\Big)\Big], \quad (3)$$

where $\Delta l(m^i, n^i)$ denote the deviation of length of link mn at the i^{th} pose

$$\Delta l(m^i, n^i) = \left(\|m^i - n^i\|^2 - \|m^1 - n^1\|^2\right)^2. \quad (4)$$

The number of terms in Equation 3 is $2(N-1) + 3(N-1)(K-1)$, whereas there are $6K+2$ variables. For $K = 1$ (i.e., four-bar linkage), there are exact solutions that yield zero error for up to five poses, but for more than five poses in general position, there is no solution that yields zero error [SR83, MS11]. For $K > 1$, there is no exact solution for more than 3 poses. Hence, our overarching approach is to minimize Equation 3 to get the best solutions, and rely on the soft energy term E_{acc} to penalize linkages that yield a high residual (Section 6). Since Equation 3 is twice differentiable, we can use any Newton or quasi-Newton method such as BFGS to minimize it. Note that while we sample the initial joints p_k^1, q_k^1, and r_k^1 inside the user-provided linkage region, the resultant joints $\hat{p}_k^1$, $\hat{q}_k^1$, and $\hat{r}_k^1$ can end up outside this region after optimization. Nonetheless, the soft energy term E_{loc} (Section 6) will penalize such a solution proportionately. Note also that when a joint lies outside its rigid body, we add a connector between them during fabrication (Section 7.1).

The optimized joints $\hat{p}_k^1$, $\hat{q}_k^1$, and $\hat{r}_k^1$ give us a multi-bar linkage mechanism, and we can proceed with the detection of order, branch, and circuit defects as well as collisions.

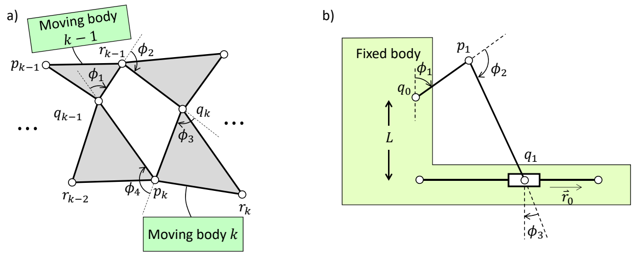

Figure 5: ***Linkage mechanism.*** *a) Four-bar loop in a multi-bar*

Defects

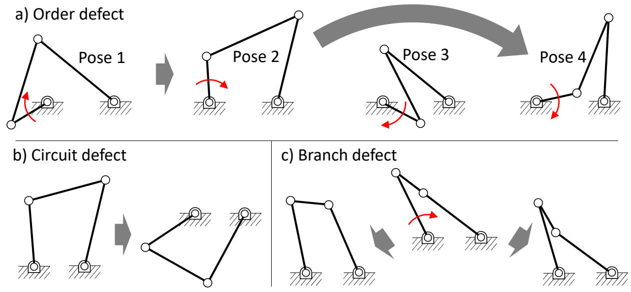

Figure 6: ***Mechanism defects.*** *a) The linkage goes from the second pose to the last pose skipping the third pose. b) The linkage cannot go from the left to the right without disassembly. c) The linkage could go from the configuration in the middle to either the left or the right when rotating the driving link (red arrow).*

4.3. RRRP slider crank

For $K = 1$, we also support the RRRP slider crank, which is another popular mechanism encountered in many every day objects. The RRRP linkage consists of a driving link and a slider. We parameterize the unit vector for the direction of the slider guide as $\vec{r}$ (Figure 5b), and the rigidity objective function becomes

$$E_{rigid}(p_1^1, q_0^1, q_1^1, \vec{r}_0) = \sum_{i=2}^{N} \Delta l(p_1^i, q_0^i) + \Big((q_1^i - q_1^1) \times \vec{r}_0\Big)^2 \quad . \quad (5)$$

The last step is to locate the two end points of the slider guide. We find these points by running a kinematic simulation of the linkage, which gives us the two extreme positions of the prismatic joint.

5. Hard Constraints

We check the candidate mechanisms against a set of hard constraints in order to reject candidates with linkage defects, poor transmission angles, or collisions. Please refer to Figure 5 for notations.

5.1. Order defect

When the mechanism traverses the poses in a different order than the input, the mechanism is said to have order defect (Figure 6a) [BC02]. Chase and Fang [CF91] address this problem for four-bar loops by analyzing the angle of the driving link in all poses (ϕ_1 in

Figure 5). We avoid this defect by ensuring all angles increase or decrease monotonically from the first pose to the last pose.

5.2. Circuit defect

A circuit defect arises when the mechanism cannot be moved between all input poses without disassembly (Figure 6b). For both 4R four-bar and slider-crank linkages, circuit defects can only occur for mechanisms where the shortest link can rotate fully with respect to a neighboring link – the so-called *Grashof* condition. In such cases, the circuit defect is avoided if each of the two angles of the link opposite to the shortest doesn't change sign across all input poses. We extend this criteria to the multi-bar linkage by evaluating the condition for each four-bar loop, $q_{k-1}r_{k-1}q_kp_k$. In the example of Figure 5a, the shortest link is $q_{k-1}r_{k-1}$, so the signs of ϕ_3 and ϕ_4 have to remain fixed. For linkages that satisfy the Grashof condition, these two angles have always the same sign, so we only need to check either one for detecting the circuit defect. Similarly, the sign of ϕ_3 has to remain fixed for the slider-crank in Figure 5b.

For each four-bar loop, the Grashof condition occurs when the following condition is satisfied:

$$|l_s| + |l_l| \leq |l_a| + |l_b|, \tag{6}$$

where $|l|$ denotes the length of link l, and l_s, l_l, l_a, and l_b denote the shortest, longest, and other two links, respectively [MS11]. The corresponding condition for the slider crank is

$$|l_s| + L \leq |l_l|, \tag{7}$$

where L is the distance between the fixed joint and the slider guide (Figure 5b).

5.3. Branch defect

A branch defect occurs when the mechanism goes through a so-called *stationary configuration*, where two adjacent links are aligned (Figure 6c). In such cases, the links can follow any of two directions as the driving link rotates, which causes driveability problems [BC02].

This defect can only occur when the link $q_{k-1}r_{k-1}$ cannot rotate fully. In such cases, the branch defect can be detected by checking the sign of the transmission angle, which is angle ϕ_3 in Figure 5. If the sign of this angle changes across the input poses, the mechanism has a branch defect.

To rotate fully, each four-bar loop has to satisfy the Grashof condition as well as the following condition [MS11]:

$$\|r_{k-1} - q_{k-1}\| + \|p_k - q_{k-1}\| < \|q_k - p_k\| + \|q_k - r_{k-1}\|. \tag{8}$$

For the slider crank, the corresponding condition is

$$\|p_1 - q_0\| < \|q_1 - p_1\| - L. \tag{9}$$

5.4. Poor transmission angle

When the transmission angle (ϕ_3 in Figure 5) is close to zero, the driving force and the torque that is transmitted to the following links vanishes, which hinders the rotation of the following links. To avoid this defect, we run a kinematic simulation and check whether the transmission angle is too close to zero during the motion. We used 10 degrees as the minimum threshold for the transmission angle. Note that when the transmission angle is zero, the branch defect occurs as well.

Thomaszewski et al. [TCG*14] avoid poor transmission angle using a soft energy term that measures the area of the triangle formed by the two adjacent links $r_{k-1}q_k$ and q_kp_k in Figure 5. Since they take an animated articulated model as input, they can evaluate this energy term for all configurations of a motion cycle. In contrast, while we considered adding a similar penalty to Equation 3 during the linkage sampling step, we found that measuring this area only at the input poses is not sufficient to avoid poor transmission angle during the entire animation.

5.5. Collision

Finally, we run a kinematic simulation from the first pose to the last one and perform discrete collision detection between the rigid bodies. We also check for collisions between the linkage and the optional user-specified safety regions. However, we do not need to check for collisions between the linkage and rigid bodies because we position the links at a different depth than the bodies (Figure 7). We also ensure that the links and joints do not collide when we generate their respective geometries, as detailed in Section 7.

Since detecting collisions requires running a kinematic simulation, which is significantly more expensive than detecting the aforementioned defects, we only test collision for the candidates that are defect free.

6. Soft Constraints

Next, we locally perturb the remaining candidates so as to refine the solution, checking for each perturbation that it does not violate the hard constraints. We define a cost function to evaluate a given linkage $\mathbf{x}$ with four soft constraints, and use this cost to guide a particle filter formulation of a stochastic global optimization [LCS16].

The soft constraints are accuracy of the traversed poses, $E_{acc}(\mathbf{x})$, smoothness of the trajectory, $E_{smooth}(\mathbf{x})$, distance of joints from the desired linkage region, $E_{loc}(\mathbf{x})$, and size of the linkage, $E_{size}(\mathbf{x})$. The first soft constraint, accuracy of the traversed poses, is defined by

$$E_{acc}(\mathbf{x}) = \frac{1}{N|\mathcal{J}_M|} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}_M} \left\| j^i - \hat{j}^i \right\|^2 \quad , \tag{10}$$

where $\mathcal{J}_M$ denotes the set of all moving joints, $|\mathcal{J}_M|$ denotes the number of moving joints, j^i denotes the input coordinates of the moving joint j at pose i, and $\hat{j}^i$ denotes the actual coordinates of the joint j at pose i. The accuracy, $E_{acc}(\mathbf{x})$, will be 0 if the linkage traverses all the input poses exactly.

The second soft constraint, smoothness of the trajectory, is approximated by the tortuosity of the trajectory between poses as follows:

$$E_{smooth}(\mathbf{x}) = \frac{1}{(N-1)|\mathcal{J}_M|} \sum_{i=1}^{N-1} \sum_{j \in \mathcal{J}_M} \frac{\int_{\text{pose } i}^{\text{pose } i+1} s_j(t)dt}{\left\| j^{i+1} - j^i \right\|} \quad , \tag{11}$$

where $s_j(t)$ denotes the speed of joint j at time t during the simulation. The summands are the ratio between the length of the motion curve between two consecutive input poses and the distance between its endpoints. This term helps the generated linkage avoid a zigzag trajectory.

The third soft constraint, location of the linkage, is evaluated as the average distance of joints at the first pose from the user-specified linkage region R,

$$E_{loc}(\mathbf{x}) = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} dist(j^1, R) \quad , \tag{12}$$

where $\mathcal{J}$ denotes the set of all joints, $|\mathcal{J}|$ denotes the number of joints, and $dist(j^1,R)$ denotes the distance of joint j at the first pose from region R.

Finally, the fourth soft constraint, size of the linkage, is measured by

$$E_{size}(\mathbf{x}) = \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} |l| \quad , \tag{13}$$

where $\mathcal{L}$ denotes the set of all links, $|\mathcal{L}|$ denotes the number of links, and $|l|$ denotes the length of link l.

Altogether, the user specifies weights to balance these four terms into a single cost function:

$$Cost(\mathbf{x}) = w_{acc}E_{acc}(\mathbf{x}) + w_{smooth}E_{smooth}(\mathbf{x}) + w_{loc}E_{loc}(\mathbf{x}) + w_{size}E_{size}(\mathbf{x}) \quad . \tag{14}$$

We use $w_{acc} = 1$, $w_{smooth} = 1$, $w_{loc} = 10$, and $w_{size} = 1$ for all the results except Figure 11 in the paper.

Using the cost function, our approach performs the particle filter optimization. It starts with the linkages that satisfied the hard constraints, $X_0 = \{\mathbf{x}_0\}$. At each iteration, the optimization perturbs the joints of each linkage $\mathbf{x}$ to get a new linkage $\mathbf{x}'$. The method uses a Gaussian distribution with a standard deviation of $0.5d$, where d is the shortest length of the bounding box of the moving body as the proposal distribution. Then, it re-checks the hard constraints in Section 5 on the perturbed linkages $\mathbf{x}'$. If a new perturbed linkage no longer satisfies all the hard-constraints, its weight is set to 0 which will cause it to be subsequently ignored. Otherwise, its weight is calculated by

$$weight(\mathbf{x}) = \exp\left[-\frac{\lambda \cdot Cost(\mathbf{x})}{\max_{\mathbf{x}_0\in X_0} Cost(\mathbf{x}_0)}\right] \quad , \tag{15}$$

where λ is the exponential decay constant. A low value of λ tends to increase the randomness of its resampling, whereas a high value of λ facilitates convergence. However, too high value of λ results in underflow of Equation 15, so we divide it by the maximum cost of the initial linkages. We use $\lambda = 20$ for all our results. Among the linkages of the previous iteration and newly added linkages, we resample 100 linkages using their weights as probabilities. Thus, even if all the new perturbed linkages have a weight of 0 due to no longer satisfying the hard-constraints, a new set of linkages can still be resampled from the previous set. After convergence, we select the lowest cost linkage as the final solution.

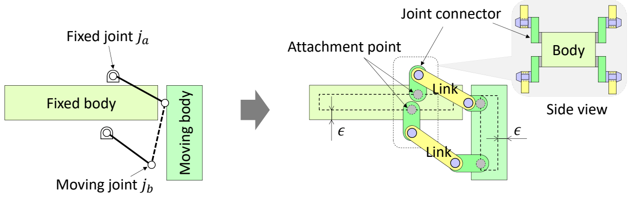

Figure 7: ***Connecting the linkage to the rigid bodies.*** *The joints of the generated linkage are automatically attached to the closest points on the rigid bodies using joint connectors.*

7. Fabrication-Ready Geometry

Given a linkage that satisfies all our previous constraints, we generate a fabrication-ready 2.5D object and assembly of links where the links are distributed at discrete depths around the fixed and moving bodies. This process involves three main steps. First, we connect each joint to its associated fixed or moving body. Second, we identify potential collisions within the linkage, from which we deduce constraints on the relative depth of the links and connectors. We solve for a depth ordering that satisfies these constraints to obtain a linkage free of collision during motion. Finally, we describe a parametric model that produces the linkage geometry given the depth of each link.

7.1. Connecting links to bodies

Our sampling procedure generates joints on and around the input rigid bodies (Section 4.1). We attach each such floating joint to the closest point of its body using a *joint connector*, which we position at an ε-distance from the boundary of the rigid body to increase robustness (Figure 7). In contrast to links, joint connectors are fixed to the rigid bodies. We fixed the dimensions of the joint connectors and links to offer a good trade-off between size and strength. Note that the joint connectors lie on the sides of the bodies to prevent collision between the links and the bodies (Figure 7, side view).

7.2. Depth-ordering

Once we have the links and joint connectors in place, we run a 2D kinematic simulation to find if they collide during the motion of the mechanism. We distinguish two collision scenarios: 1. when a link collides with the attachment point of a joint connector (Figure 8a), and 2. when a link collides with a joint (Figure 8b). These two scenarios also apply to joint connectors.

To avoid the first type of collision, the joint connector and its attachment point need to be closer to the body than the moving link. Denoting $z(c)$ the depth of the joint connector with respect to the body, and $z(l)$ the depth of the moving link, we obtain the ordering inequality

$$z(c) < z(l). \tag{16}$$

To avoid the second type of collision, the moving link needs to be in front or behind all the links and joint connectors connected to the

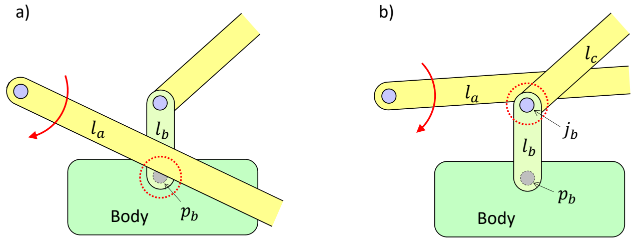

Figure 8: ***Two cases of collision between links.*** *During the kinematic simulation, two types of collisions in 2D projected space are recorded: a) link l_a collides with p_b, the attachment point of joint connector l_b. b) link l_a collides with joint j_b.*

colliding joint. Denoting $\mathcal{L}_j$ this set of links and joint connectors, we obtain the disjunctive ordering inequality

$$z(l) < \min_{l_j \in \mathcal{L}_j} z(l_j) \bigvee z(l) > \max_{l_j \in \mathcal{L}_j} z(l_j). \tag{17}$$

In addition to these two constraints, we also need to prevent adjacent links or joint connectors to be at the same depth due to our linkage geometry design (Figure 9). Denoting l_a and l_b two links or connectors sharing a joint, we express this constraint as

$$z(l_a) \neq z(l_b), \tag{18}$$

which can be seen as a disjunctive inequality

$$z(l_a) < z(l_b) \bigvee z(l_a) > z(l_b). \tag{19}$$

The same constraint also applies when two links l_a and l_b intersect in the first pose.

Finally, we solve for a depth ordering of the links and joint connectors that satisfies all the constraints listed above. Since several solutions may exist, we favor the one that occupies the least space, i.e. with the smallest depth complexity $\max_{l \in \mathcal{L}} z(l)$. Coros et al. [CTN*13] tackle a similar ordering problem, which they solve with an off-the-shelf Constraint Satisfaction Problem (CSP) solver. We follow their approach using Gecode [Gec18], which finds a solution in a few milliseconds for our problem. However, some candidate mechanisms may not admit any solution, in which case we reject those candidates.

7.3. 3D geometry

Given a valid depth ordering of links and joint connectors, our system automatically generates the geometry of the mechanism based on a parametric linkage model. Figure 9a illustrates the parameters of this model on one side of an object, the opposite side being obtained by symmetry. We define the distance between two components of depth order $z(l_a)$ and $z(l_b)$ as

$$height\big(z(l_a) - z(l_b)\big) = \big(z(l_a) - z(l_b)\big)\big(LD + JD + 2GP\big) - LD \quad , \tag{20}$$

where LD is the height of a link, JD is the height of a joint cap, and GP is the gap size between components of adjacent depth orders. The distance between a joint connector and the body is calculated in a similar manner. We adjusted the values of LD, JD and GP based on the scale and precision of our 3D printer.

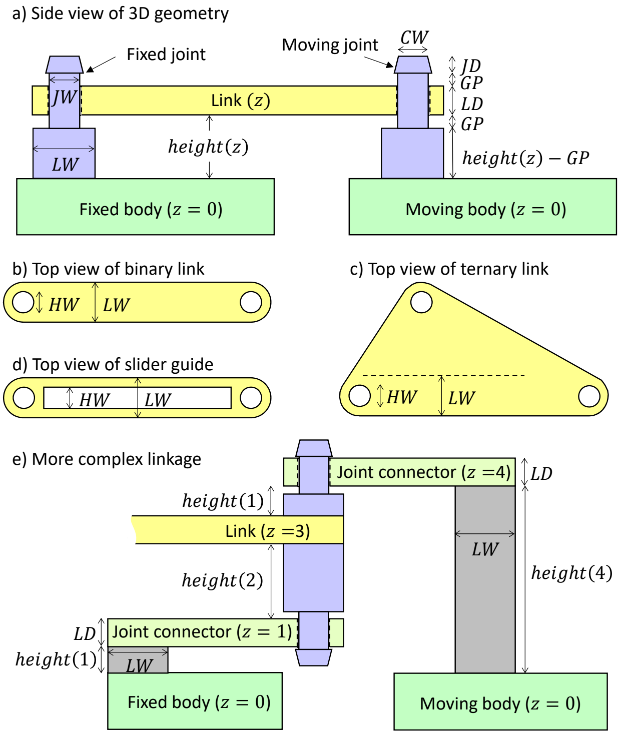

Figure 9: ***Parameterized linkage geometry.*** *a) The height of the joints is computed from the depth of the link, z. b) Top view of a binary link. c) Top view of a ternary link. d) Top view of a slider guide. e) In the case where three links are connected via a joint, the joint should be attached to the link that is located in the middle, and other links have a hole at the corresponding position.*

In the case where three links are connected via a joint, the joint is attached to the link that is located in the middle, and other links have a hole at the corresponding position (Figure 9). The height of the joint can be determined by Equation 20.

8. Results

We implemented our approach on an i7-based PC workstation with 12 GB of memory and NVidia GTX980 graphics card. Our tool was implemented in C++ using OpenGL/GLSL. For minimizing Equations 3 and 5, we use the BFGS algorithm implemented in dlib C++ library [Kin09]. For the kinematic simulation, we implemented our custom simulator following the symbolic reconstruction approach [BBJP12]. We found that symbolic reconstruction is faster than solving the so-called *position loop equations*, which involve a lot of trigonometric calculations [MS11].

8.1. Generated mechanical objects

Figure 10 shows multiple mechanical objects created using our sketching tool. Most of the examples require a complex motion.

For instance, the car roof, sofa bed, and folding table need a particular trajectory to avoid collisions. Figure 1 shows a retractable car roof, for which the user specified three key poses of the two roof parts. Also, the power shovel, bulldozer, and garbage truck examples in Figure 10 use three poses to control the orientation of the moving body during the motion. For instance, the garbage truck keeps its waste container upright until right before it reaches the top of the truck. Please refer to the accompanying video for animations of these mechanisms.

Our approach allows the user to interactively obtain different solutions by changing the weights for the soft-terms, the accuracy, smoothness, and size of the linkage (Figure 11). Since the input poses might not be accurate, the user can balance the accuracy and other soft-terms to obtain more desired linkage.

8.2. 3D printing

The output of our approach is a 3D geometry that consists of the main rigid bodies that the user sketched and links that are generated by our system. Figure 12 shows the printed parts of the wall bed, wall chair, sofa bed, garbage truck, and folding table. The physical objects can be assembled without any additional bolts and screws. It took less than a minute to assemble the fabricated parts for each result in the paper. Please refer to the accompanying video for the actual assembly of some of the printed objects.

8.3. Performance

Table 1 shows the computation time for the examples in Figures 1 and 10. For the initial sampling, $100,000$ samples are randomly generated while most of them are quickly discarded due to the order, circuit, and branch defects (Third column of Table 1), which helps reduce the computation time of the kinematic simulation that is performed to check the collision only for the remaining solutions. Then, 20 iterations of particle filter optimization (PFO) are performed. For both initial sampling and particle filter optimization, we use 8 threads to parallelize the approach. While running a kinematic simulation to detect collisions is costly, our sampling method effectively discards most of the candidates that do not satisfy the hard-constraints, which helps reduce the computation time. The kinematic simulation can run even faster by increasing the step size of the driving crank. However, too big of a step size may cause skips when the moving body collides with the fixed body; thus, we use a step size of 1 degree for the driving link to avoid such an issue during simulation.

Table 1: ***The computation time for generating the mechanism for the examples in Figures 1 and 10.*** *For sofa bed and bulldozer, the slider crank mechanism is selected, whereas multi-bar linkages are used for the other examples.*

	Col. chk / Sampling (sec)	#Solutions		PFO (sec)
		before col. chk	after	
Car ($K=2$)	0.33 / 2.50	1,781	35	0.30
Wall bed ($K=2$)	0.09 / 2.01	3,630	32	0.37
Wall chair ($K=2$)	0.17 / 2.03	2,112	186	0.31
Car ($K=3$)	0.33 / 4.12	451	5	0.24
Power shovel ($K=3$)	0.37 / 4.68	77	10	0.33
Sofa bed (slider crank)	0.03 / 0.44	213	49	0.07
Bulldozer (slider crank)	0.06 / 0.61	133	119	0.08
Garbage truck ($K=1$)	0.21 / 0.46	4,267	606	0.80
Folding table ($K=1$)	0.05 / 0.20	909	225	0.21

8.4. Gaussian noise to input poses

Figure 13a shows the effect of the Gaussian noise σ added to the input poses. The cost of the best linkage in terms of the soft constraints decreases as σ increases. For the four-bar linkage, the exact solution that yields $E_{acc}(\mathbf{x}) = 0$ exists for up to five poses without any perturbation to the input poses (i.e., σ=0). There are only a few solutions for five input poses (Top left box). However, the exact solution may not be optimal in terms of other constraints, smoothness of the trajectory, and the location and size of the linkage. By increasing the Gaussian noise σ added to the input poses, more variation of solutions can be obtained (blue dots in Figure 13), which allows us to get a better linkage mechanism that does not have any defect and approximately traverses the input poses without any collision.

A similar improvement can be observed even if we increase the number of moving bodies from $K = 1$ to 3 (Figure 13b). However, increasing the number of moving bodies also increases the number of unknowns and constraints, which makes the number of valid solutions decrease significantly. Increasing the number of initial samples could be a remedy, at the cost of increasing computation time.

8.5. Particle filter optimization

Figure 14 compares the best cost found by our approach and by a degraded version without particle filter optimization. For a given computational budget, particle filter optimization allows our method to find better configurations. Also, please see the supplemental materials for the analysis of the convergence of the cost distribution.

8.6. Comparison to previous work

Figure 15 provides a comparison to the linkage synthesis approach of Megaro et al. [MTG*14]. The original method by Megaro et al. generates two types of links – connectors (Fig. 3 in their paper), and motion propagators and trimmers (Fig. 4 and 5 in their paper). Connectors connect the rigid bodies such that they traverse the two specified poses, while motion propagators and trimmers propagate motion from a single actuator to all rigid bodies. While Megaro et al. avoid branch defect (and the related poor transmission angle)

Results

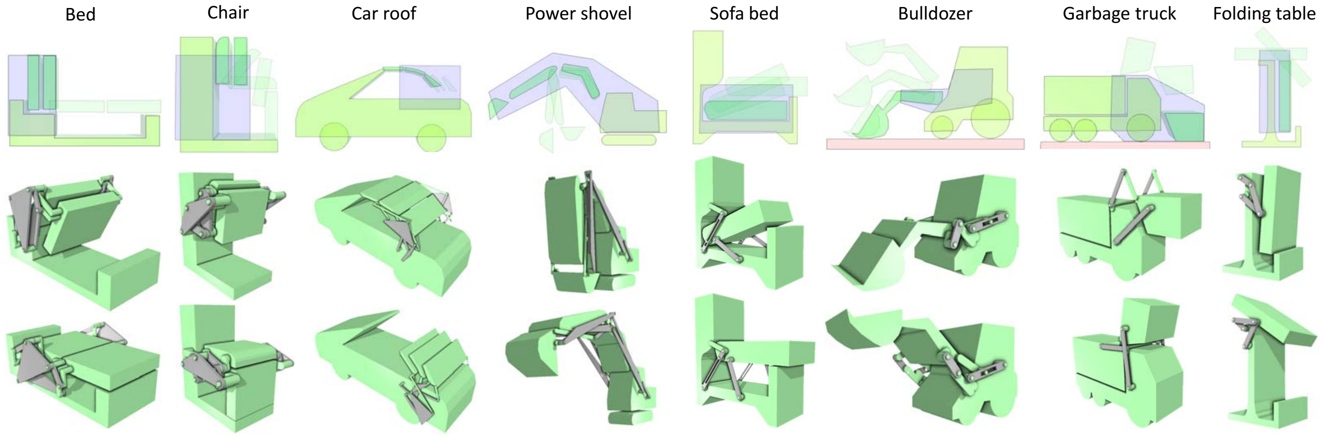

Figure 10: ***Generated dynamic objects.*** *Each column shows a dynamic object created with our system. The top row shows the input sketches including the poses and desired region for the initial location of the linkage. The middle and bottom rows show the generated 3D geometry that consists of the designed bodies and generated linkage. The first two examples have two moving bodies, whereas the third and forth have*

sm, and the last two columns use a 4R four-bar linkage. Please

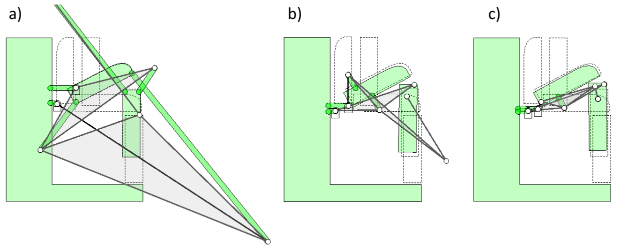

Figure 11: ***Alternative solutions.*** *The user can interactively obtain different solutions by changing the weights for the soft-terms. a) When only accuracy is taken into account, the moving bodies traverse the input poses very accurately, but the linkage might become unnecessarily large. b) More smooth trajectory can be achieved by sacrificing the accuracy and size, while c) a smaller size of linkage can be achieved by sacrificing the accuracy and smoothness.*

Figure Y: Alternative solutions. The user can interactively obtain different solutions by changing the weights for the soft-terms. a) When only accuracy is taken into account, the moving bodies traverse the input poses very accurately, but the linkage might become unnecessarily large. b) More smooth trajectory can be achieved by sacrificing the accuracy and size, while c) a smaller size of linkage can be achieved by sacrificing the accuracy and smoothness.

for motion propagators and trimmers (Sec. 5 of their paper), they do not prevent such defects when sampling their connectors (Sec. 4.1 of their paper). Our comparison focuses on these connectors since they are the ones defining the trajectory of the moving body, and may cause collisions (Figure 15a-b), undesired trajectory (Figure 15c), and some defects (Figure 15d).

Figure 16a shows a complex object for which two extreme poses are not enough to achieve the desired trajectory. Unlike the method by Megaro et al., our approach allows users to refine the trajectory by adding intermediate poses (Figure 16b).

8.7. User study

We conducted a user study to evaluate the benefits of our automatic approach over manual linkage design. We recruited eight voluntary graduate-school-age participants without any experience in linkage design. After letting the users have time to play with our system, we asked each participant to perform two specific design tasks (Figure 17). In the first task, two poses of a moving body are provided, and the subject is required to manually design a four-bar linkage that moves the moving body from the first pose to the second pose. In the second task, three poses are provided, and the shape of the fixed body adds more chance of collision for the moving body, which makes the task more difficult. The participants can claim that the task is completed if the moving body approximately traverses the input poses. Also, they can give up the task if they find it too difficult.

Table 2: ***Summary of the manual design task*** *The average of E_{acc} is shown in terms of d, where d is the shortest length of the bounding box of the moving body.*

	Avg. design time (min)	Give up (%)	Avg. E_{acc}	Our E_{acc}
Task 1	3.9	0	$0.23d$	$0.0000075d$
Task 2	14.6	37.5	$0.98d$	$0.00045d$

Table 2 summarizes the results of this user study, and Figure 17 shows some of the designed linkages. Please refer to the supplemental materials for all the designed linkages by the participants. While participants managed to design a reasonable linkage in a few minutes for the simple task, three did not complete the second task, which took 15 minutes on average for the five remaining participants. In addition, the resulting linkages do not traverse the input poses precisely (E_{acc} in Table 2). In contrast, our system produces compact mechanisms that accurately capture the target motion for both tasks.

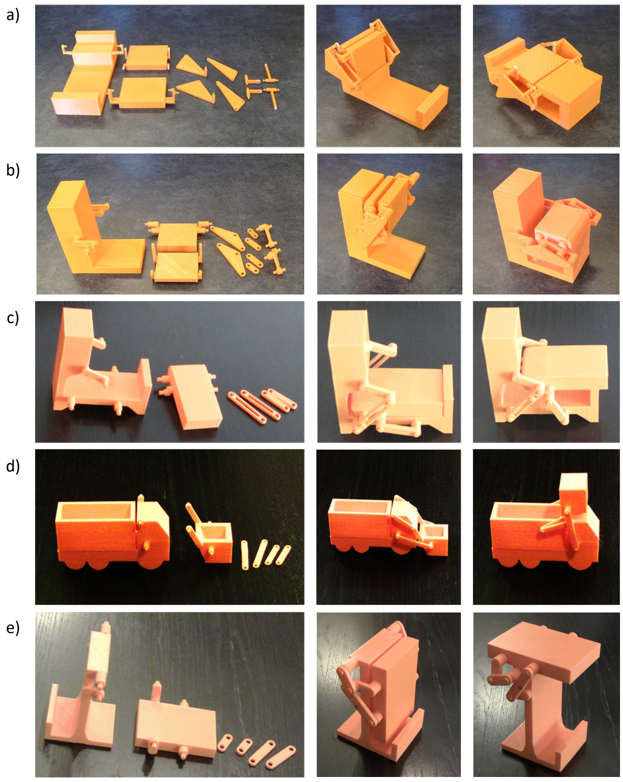

Figure 12: ***3D printed examples.*** *The output 3D geometry of a) wall bed, b) wall chair, c) sofa bed, d) garbage truck, and e) folding table were printed with a desktop 3D printer. The printed parts can be assembled readily without additional bolts and nuts.*

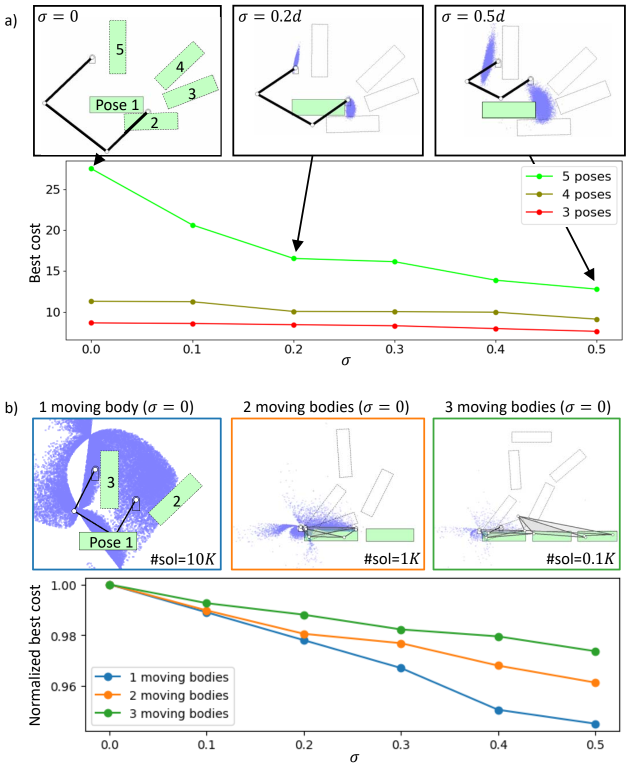

Figure 13: ***Solution distribution analysis.*** *a) By increasing the Gaussian noise σ for the input poses, more variation of solutions can be obtained, which increases the chance to find better solutions in terms of our soft constraints. The top row shows the best linkage obtained for the perturbation noise $\sigma = 0$, $0.2d$, and $0.5d$, where d is the shortest length of the moving body, and the solution distribution for the fixed joints (blue dots). Notice that there are only a few solutions for five poses without any perturbation (i.e., σ=0). While the linkages for $\sigma = 0.2d$ and $0.5d$ do not traverse the input poses exactly, the size of the obtained linkage becomes smaller to achieve a lower cost. Please refer to the supplemental materials for the solution distribution for 3 poses and 4 poses. b) Increasing the number of moving bodies adds more constraints, which reduces the valid solutions. The number in each box shows the number of valid solutions. Adding the Gaussian noise still improves the obtained linkage in terms of the cost.*

9. Conclusions and Future Work

We proposed an interactive system to quickly design a mechanical object that produces a desired motion. Users only interact with the system by sketching disconnected fixed and moving bodies, while all the burden of linkage design is handled by our automatic approach. Guided by the long history of four-bar, six-bar and multi-bar linkage studies, we efficiently generate planar linkages that avoid common defects. Collision is also taken into account by running a kinematic simulation and by automatically ordering links in depth. As such, our system greatly accelerates prototyping of physical, dynamic objects.

While our generalization of four-bar, six-bar, and eight-bar linkages to multi-bar linkages can generate many everyday objects, there are still some complex designs that cannot be effectively represented by our mechanism as illustrated in Figure 18. The expressiveness of our system could be expanded by supporting other types of linkages with various types of joints, such as gear, cam, and screw. However, several ingredients of our approach are linkage-dependent. First, the rigidity energies E_{rigid} (Equation 3 and 5) used to generate candidate linkages are specific to our custom multi-bar linkages. Similarly, our parametric linkage model would need to be extended to fabricate other types of joints. Additional hard constraints may also be needed to cover the defects of other linkages. Finally, more complex linkages have additional components, which increases the risk of collision between components, and the computation time for finding their valid depth ordering.

The thickness of the links and the size of the joints in our results are based on the precision of our desktop 3D printer, which results in big mechanisms for the small prototypes we fabricated. The relative size of the mechanism with respect to the overall object would reduce if we used our approach to design bigger objects, or if we used more sturdy materials. However, our approach does not consider the driving force when finding the best mechanism. It would

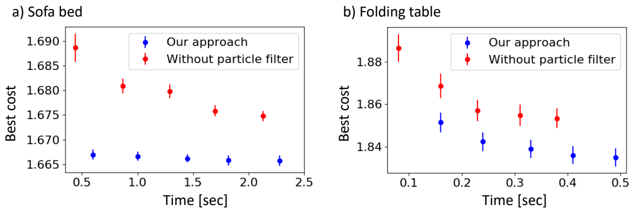

Figure 14: ***Effect of particle filter optimization.*** *The blue points represent the best cost by our approach, whereas the red points represent the best cost obtained without particle filter optimization. For both, the points represent the mean values of 100 trials while the vertical bars show the 95% confidence interval.*

haCra [Megaro et al. 2014]

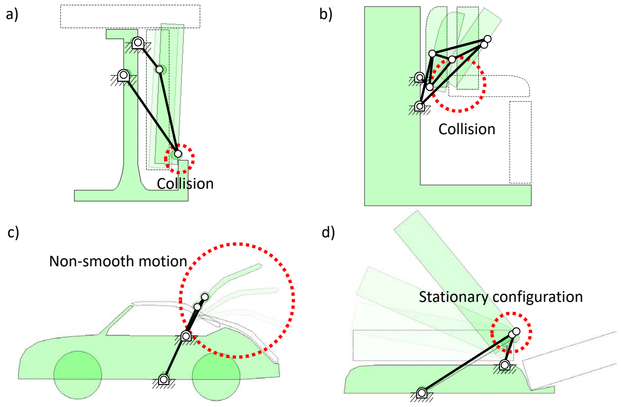

Figure 15: ***The results by [MTG*14].*** *The linkage synthesis approach of Megaro et al. [MTG*14] does not consider any defect, collision, and the smoothness of the trajectory. As a result, the generated connectors may cause a-b) collisions, c) undesired trajectory, and d) some defects.*

be desired to minimize the driving force, especially when the size of the mechanism is scaled up. Supporting linkages for 3D motion is also an exciting direction of future work, although it raises numerous challenges for collision detection and linkage fabrication.

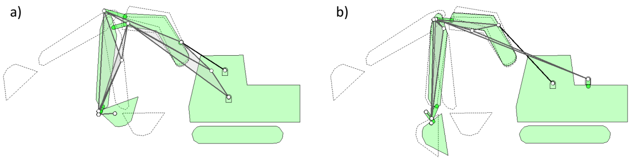

Figure X. **Motion controllability.** Two poses may not be enough to control the motion. a) Only two poses result in an undesired intermediate pose for the dipper to dig. b) By adding an intermediate pose, the user can obtain more desired motion.

Figure 16: ***Motion controllability.*** *Two poses may not be enough to achieve a desired motion. a) Using only two poses, the dipper shovel does not dig into the ground. b) Adding an intermediate pose allows the user to produce the proper motion.*

User Study

Xiaowei Andres Seunghoon

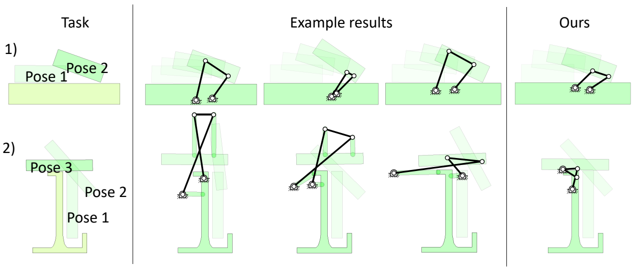

Figure 17: ***User study.*** *The subjects were asked to do two tasks, 1) easy and 2) difficult ones. While most of the subjects could successfully find a valid solution that does not cause any collision from the first pose to the last pose, there is a significant error in the trajectory. Please see the accompanying video for the actual motion of*

Limitations

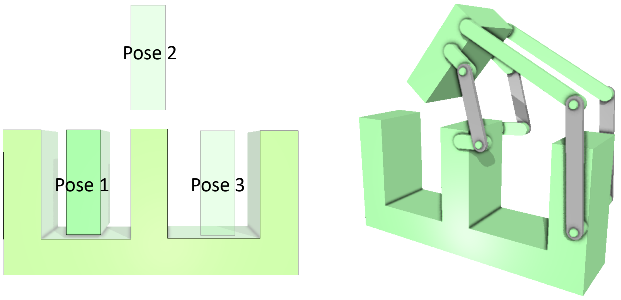

Figure 18: ***Limitations.*** *Our approach cannot support all possible poses, such as this example; as future work we could incorporate more complex linkage types.*

Acknowledgements

We would like to thank our user study participants for their time and feedbacks, and the anonymous reviewers for their constructive suggestions. This research was partially funded by the ERC starting grant D^3 (ERC-2016-STG 714221), NSF CBET 1250232, IIS 1302172, CHS 1816514, and by research and software donations from Adobe.

References

[BBJP12] Bächer M., Bickel B., James D. L., Pfister H.: Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. 31*, 4 (2012), 47:1–47:9. 8

[BC02] Balli S., Chand S.: Defects in link mechanisms and solution rectification. *Mechanism and Machine Theory 37*, 9 (2002), 851 – 876. 2, 5, 6

[BCT15] Bächer M., Coros S., Thomaszewski B.: Linkedit: Interactive linkage editing using symbolic kinematics. *ACM Trans. Graph. 34*, 4 (2015), 99:1–99:8. 3

[BD09] Bulatovic R. R., Dordevic S. R.: On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations. *Mechanism and Machine Theory 44*, 1 (2009), 235 – 246. 3

[Bur86] Burmester L.: *Lehrbuch der Kinematik*. Verlag Von Arthur Felix, 1886. 2

[CF91] Chase T. R., Fang W. E.: Order rectification for complex number based burmester curves. *Journal of Mechanical Design - Transactions of the ASME 113*, 3 (1991), 239–247. 2, 5

[CKE13] Chase T. R., Kinzel G. L., Erdman A. G.: Computer aided mechanism synthesis: A historical perspective. *Advances in Mechanisms, Robotics and Design Education and Research* (2013), 17–33. 2

[CLM*13] Ceylan D., Li W., Mitra N. J., Agrawala M., Pauly M.: Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph. 32*, 6 (2013), 186:1–186:11. 3

[CM93] Chase T. R., Mirth J. A.: Circuits and branches of single-degree-of-freedom planar linkages. *Journal of Mechanical Design - Transactions of the ASME 115*, 2 (1993), 223–230. 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. Graph. 32*, 4 (2013), 83:1–83:12. 3, 8

[Fil72] Filemon E.: Useful ranges of centerpoint curves for design of crank-and-rocker linkages. *Mechanism and Machine Theory 7*, 1 (1972), 47 – 53. 2

[Gec18] Gecode: Gecode: Generic constraint development environment, 2018. Available from `http://www.gecode.org`. 8

[GLC*16] Goulet V., Li W., Cheong H., Iorio F., Quimper C.: *Four-Bar Linkage Synthesis Using Non-convex Optimization*. Springer International Publishing, 2016, pp. 618–635. 3

[GM12] Gogate G. R., Matekar S. B.: Optimum synthesis of motion generating four-bar mechanisms using alternate error functions. *Mechanism and Machine Theory 54* (2012), 41 – 61. 3

[HD64] Hartenberg R. S., Denavit J.: *Kinematic Synthesis of Linkages*. McGraw-Hill, 1964. 2

[Kin09] King D. E.: Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research 10* (2009), 1755–1758. 8

[KLY*14] Koo B., Li W., Yao J., Agrawala M., Mitra N. J.: Creating works-like prototypes of mechanical objects. *ACM Trans. Graph. 33*, 6 (2014), 217:1–217:9. 3

[KN17] Kafash S. H., Nahvi A.: Optimal synthesis of four-bar path generator linkages using circular proximity function. *Mechanism and Machine Theory 115* (2017), 18 – 34. 3

[LCS16] Liu B., Cheng S., Shi Y.: Particle filter optimization: A brief introduction. In *Advances in Swarm Intelligence* (2016), Springer International Publishing, pp. 95–104. 2, 6

[MA08] Martínez-Alfaro H.: Four-bar mechanism synthesis for n desired path points using simulated annealing. *Advances in Metaheuristics for Hard Optimization* (2008), 23–37. 3

[MC95] Mirth J. A., Chase T. R.: Circuit rectification for four precision position synthesis of four-bar and watt six-bar linkages. *Journal of Mechanical Design 117*, 4 (1995), 612–619. 2

[MS11] McCarthy J. M., Soh G. M.: *Geometric Design of Linkages*. Springer-Verlag, 2011. 2, 3, 4, 5, 6, 8

[MTG*14] Megaro V., Thomaszewski B., Gauge D., Grinspun E., Coros S., Gross M.: Chacra: An interactive design system for rapid character crafting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), SCA '14, pp. 123–130. 3, 4, 9, 12

[MZB*17] Megaro V., Zehnder J., Bächer M., Coros S., Gross M., Thomaszewski B.: A computational design tool for compliant mechanisms. *ACM Trans. Graph. 36*, 4 (2017), 82:1–82:12. 3

[PM14] Plecnik M. M., McCarthy J. M.: Numerical synthesis of six-bar linkages for mechanical computation. *Journal of Mechanisms and Robotics 6*, 3 (2014). 2

[PMW14] Plecnik M., McCarthy J. M., Wampler C. W.: Kinematic synthesis of a watt i six-bar linkage for body guidance. *Advances in Robot Kinematics* (2014), 317–325. 2

[San59] Sandor G. N.: *A General Complex-number Method for Plane Kinematic Synthesis with Applications*. Columbia University, 1959. 2

[SE84] Sandor G. N., Erdman A. G.: *Advanced Mechanism Design: Analysis and Synthesis*. Prentice-Hall, Inc., 1984. 2

[SR67] Suh C. H., Radcliffe C. W.: Synthesis of plane linkages with use of the displacement matrix. *Journal of Engineering for Industry 89*, 2 (1967). 2

[SR83] Suh C., Radcliffe C. W.: *Kinematics and Mechanisms Design*. Krieger Pub Co, 1983. 5

[TCG*14] Thomaszewski B., Coros S., Gauge D., Megaro V., Grinspun E., Gross M.: Computational design of linkage-based characters. *ACM Trans. Graph. 33*, 4 (2014), 64:1–64:9. 3, 6

[Wal76] Waldron K. J.: Elimination of the branch problem in graphical burmester mechanism synthesis for four finitely separated positions. *Journal of Engineering for Industry 98*, 1 (1976), 176–182. 2

[ZSC16] Zheng C., Sun T., Chen X.: Deployable 3d linkages with collision avoidance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), Eurographics Association, pp. 179–188. 3

[ZXS*12] Zhu L., Xu W., Snyder J., Liu Y., Wang G., Guo B.: Motion-guided mechanical toy modeling. *ACM Trans. Graph. 31*, 6 (2012), 127:1–127:10. 3