Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

An Interactive Design System of Free-Formed Bamboo-Copters

Morihiro Nakamura, Yuki Koyama, Daisuke Sakamoto and Takeo Igarashi

The University of Tokyo

Figure 1: *Using our interactive design system, users can design free-formed, even asymmetric, bamboo-copters that can be easily fabricated and can successfully fly.*

Abstract

We present an interactive design system for designing free-formed bamboo-copters, where novices can easily design free-formed, even asymmetric bamboo-copters that successfully fly. The designed bamboo-copters can be fabricated using digital fabrication equipment, such as a laser cutter. Our system provides two useful functions for facilitating this design activity. First, it visualizes a simulated flight trajectory of the current bamboo-copter design, which is updated in real time during the user's editing. Second, it provides an optimization function that automatically tweaks the current bamboo-copter design such that the spin quality—how stably it spins—and the flight quality—how high and long it flies—are enhanced. To enable these functions, we present non-trivial extensions over existing techniques for designing free-formed model airplanes [UKSI14], including a wing discretization method tailored to free-formed bamboo-copters and an optimization scheme for achieving stable bamboo-copters considering both spin and flight qualities.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques. I.3.8 [Computer Graphics]: Applications—. I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling.

1. Introduction

The *bamboo-copter*, also known as *bamboo dragonfly* or *taketombo*, is a traditional toy that has several rotor blades, or rotationally symmetrically placed wings, and can fly by spinning. Since the creation of the bamboo-copter in about 400 BC in China [Lei06], their movements have caused fascination. Typical bamboo-copters, however, have a very limited variation in wing shape. As shown in Figure 2, most bamboo-copters have two wings that have a mostly rectangular shape. One reason for this is that, traditionally, these wings are crafted from one bamboo fragment, so that the design is constrained. However, with the popularization of digital fabrication technologies, these constraints on craft techniques have been minimized. Nevertheless, designing creative and functional bamboo-copters is still extremely challenging; a designer has to consider both the *spin quality*—how stably it spins—and the *flight quality*—how high and long it flies—in the design plan.

We present an interactive design system for designing bamboo-

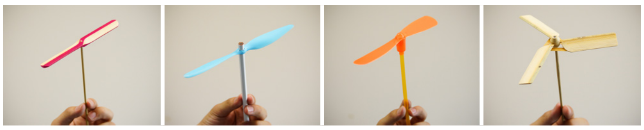


Figure 2: *Bamboo-copters that we bought from online shops. We chose different types so that they are as diverse as possible. Note that they still have very similar wings.*

copters with free-formed wing shapes (see Figure 1). Our system enables users to design creative, even asymmetric, but still functional bamboo-copters that can be fabricated using digital fabrication equipment, such as a CNC (computerized numerical control) laser cutter and a standard 3D printer. Users can interactively design free-formed planar wings by drawing their shapes and manipulate how wings are attached to the stick. Our system provides two useful functions for facilitating this design activity. First, it visualizes a simulated flight trajectory of the current bamboo-copter design, which is updated interactively during the user's editing. Second, it provides an optimization function that automatically tweaks the current bamboo-copter design such that the spin and flight qualities are enhanced.

Our target objective, a bamboo-copter design, has many similarities to model airplane (or glider) design. Based on the previous work on a model airplane design system presented by Umetani et al. [UKSI14], we offer non-trivial extensions over it to adapt their techniques for the design of a bamboo-copter. For simulating flight trajectories of bamboo-copters, we take an *annular* wing discretization approach that is different from the previous one. To obtain physical coefficients related to aerodynamics, we also take a data-driven approach following the work of Umetani et al., but use a different formulation specialized for bamboo-copters. The formulation for the optimization function is also different; while the previous method considers only the flight quality, our method incorporates the spin quality into the objective function, which is specifically important in bamboo-copter design. For this, we extend the insights from a study on designing spinnable objects presented by Bächer et al. [BWBSH14].

The proposed system is immediately useful for those who want to design their own bamboo-copters. Toy companies may hire a designer to design a sophisticated bamboo-copter using the system and sell it as a product. A hobbyist may design a customized copter and play with it, or give it to somebody else as a present. Our formulation of flight simulation and optimization might be useful for the design of artifacts with spinning components other than bamboo-copters, such as drones in the future.

Figure 3 illustrates the overview of our approach. In summary, this paper offers the following contributions:

Novel application. We present a novel design system specialized for free-formed bamboo-copters. To our knowledge, this is the first attempt to enable novices to design free-formed bamboo-copters with computational support.

Data-driven flight simulation. Our data-driven flight simulation framework closely follows that of [UKSI14], but we extend the data-acquisition and data-fitting procedure to work for bamboo-copter flights driven by rotary wing motion.

Design optimization. We formulate how a bamboo-copter can be optimized such that both the spin and flight qualities are enhanced. Our optimization workflow first applies spin axis alignment and then optimizes spin and flight qualities.

2. Related Work

Aerodynamics. Research on aerodynamics has a long history. Here, we focus on how aerodynamics is discussed in the field of computer graphics. Motion of flapping birds is generated [WP03] and controlled [JWL*13] based on aerodynamics. For designing flyable things, Umetani et al. [UKSI14] and Martin et al. [MUB15] enabled to design creative model airplanes and kites respectively based on acquired aerodynamic properties. We build on these works and present how free-formed bamboo-copters can be simulated and optimized.

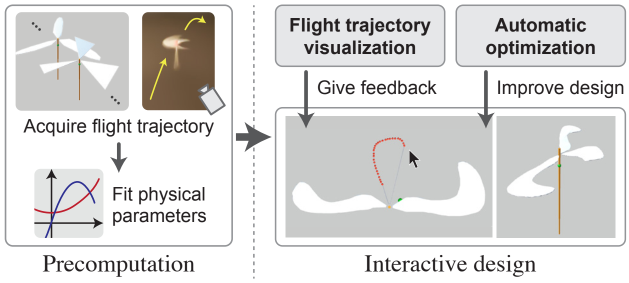


Figure 3: *Overview. First, we fabricate and fly many bamboo-copters to acquire flight trajectories. Using the acquired data, we fit physical parameters which are necessary for simulating and optimizing bamboo-copters. In interactive design session, our system allows users to design free-formed, flyable bamboo-copters by visualizing useful information such as a simulated flight trajectory, and also by providing an automatic optimization function.*

Fabrication-Oriented Functional Design. Researchers have investigated computational methods for designing functional, fabricable objects, in which various functionalities have been formulated, including standing stability [PWLSH13], rotational stability [BWBSH14], floating stability [WW16], hold-ability of 3D-printed connectors [KSS*15], structural strength [SVB*12, LSZ*14] and deformation behavior of 3D-printed objects [PZM*15, SBR*15]. These works utilize optimization frameworks to maximize or ensure functionality of designed objects. In this work, we present an optimization framework for improving the functionality of a bamboo-copter, *i.e.*, spin and flight qualities.

To let end-users design creative yet functional objects, interactive design systems with physically-based feedback have been investigated. These include systems for designing, for example, furniture [SLMI11, UIM12], metallophones [UTMI11], model airplanes [UKSI14], and kites [MUB15]. Our system predicts how bamboo-copters fly and provides interactive feedback based on it.

Measurement-Based Parameter Fitting. Pai et al. [PDJ*01] presented a method for simulating realistic physical behaviors by fitting model parameters from measurements. Following this seminal work, data-driven physics models have been investigated in various ways [BBO*09, MBT*12, UKSI14, MUB15]. We estimate physical parameters that are necessary for flight simulation of bamboo-copters from captured actual flight data. We specifically follow Umetani et al.'s work [UKSI14], where they fitted aerodynamic parameters for model airplanes. Unlike model airplanes, aerodynamic forces are mainly genarated by spin motion in bamboo-copters. To take angular velocities into account, we construct a simple capture environment and a formulation for bamboo-copters.

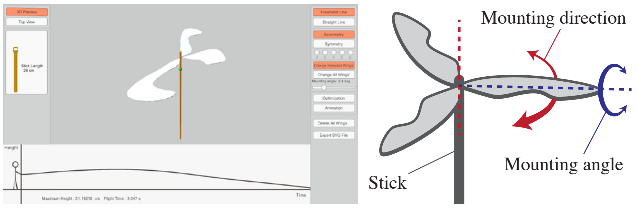


Figure 4: *(Left) Screen capture of our system. (Right) Mounting angle and direction.*

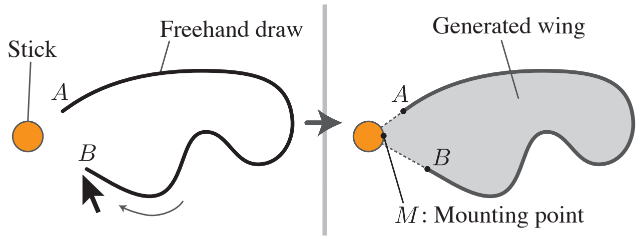


Figure 5: *User input. From the top view, the user draws an stroke from A to B that represents the wing shape. The system then generates a wing by filling A, B, and the center of the stick.*

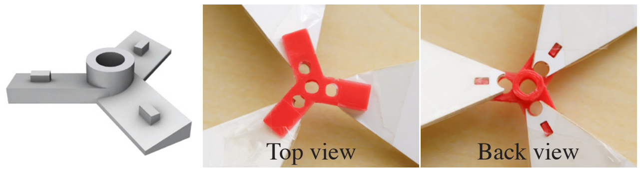


Figure 6: *A 3D-printed connector for assembling a bamboo-copter, which enables accurate control of mounting angles and positions of wings. Note that three additional holes in these photos are added to use the launch pad.*

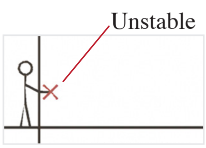
3. System Overview

Figure 4 (Left) shows a screen capture of our system. Initially, a default-length stick without any wing appears in the system. The user starts by drawing wing shapes from the top view. Once the user finishes drawing all the desired wings (usually 2–4 wings), the user adjusts two rotational configuration parameters of the wings: the *mounting angles* and the *mounting directions* (see Figure 4 (Right)). Our system currently does not support the other rotational control of wings. The user can also adjust the length of the stick, and the vertical position of the wings on the stick.

Our system visualizes the center of mass of the bamboo-copters as a green dot. In addition, in the left-bottom corner of the screen (see Figure 4 (Left)), the system shows the result of flight simulation. It is updated every time the user manipulates the wings and the stick. The user can also see the simulation result as an animation in the main panel by clicking the "Animation" button. In addition, our system can optimize configuration of the bamboo-copter in order to fly more stably and higher. Finally, our system exports the resulting wing patterns as a SVG file so that the user can fabricate the wings using a CNC laser cutting machine. The system also exports 3D geometry of a joint that connects the wings to the stick, so that the user can fabricate the joint using a 3D printer.

Wing Generation. Figure 5 shows how to generate a wing. First, from the top view, the user draws a freehand stroke that represents the contour of the intended wing. Suppose that the line starts from A and ends at B. The system generates a wing by connecting A and B to the stick. The *mounting point* M of this wing is defined as the middle point between the intersections of the wing contour and the stick. Note that our current system can handle only planar wings for simplicity in fabrication and aerodynamic analysis. It is a future work to consider non-planar, 3-dimensional wing shapes (*e.g.,* twisted wings) to enable more creative designs.

Flight Trajectory Visualization. The system visualizes the simulated flight trajectory in a graph view. The abscissa of the graph is the time axis and the ordinate of the graph is the height axis. This simulation starts with the initial angular velocity 160 rad/s and the pose of 5-degrees tilted from the vertical axis, from the 1.5 meters high. We slightly tilt it from the vertical axis because bamboo-copters are launched by hand and it is difficult to keep it perfectly vertical in practice. If a bamboo-copter goes to an unstable flight situation (when the stick inclines over 30 degrees), then our system displays a red X mark on the trajectory to the user, as shown in the inset figure.



Automatic Optimization of the Current Design. Once the user pushes the "Optimization" button on the interface, the system optimizes the configuration of the bamboo-copter to maximize spin quality and flight quality. By this optimization, the scale, mounting position, mounting direction, mounting angle of each blade, as well as the vertical position of the wings and the length of the stick, are optimized. This computation usually takes 1 or 2 minutes.

Bamboo-Copter Fabrication. The user can fabricate bamboo-copters using sticks, joints, and lightweight boards. In this paper, 4-mm-diameter wood sticks and 1-mm-thickness foam core boards are used. Foam core boards are chosen because they are not only lightweight and rigid, but also easy to cut into free-formed wing shapes by a CNC laser cutter. Joints are printed by using a standard FDM-based 3D printer with ABS, and used to fix the wings to the stick and to set the mounting angles and the mounting directions precisely (see Figure 6). We fix them using Sellotape. In simulation, we simply ignore the mass of joints and tape since they have little effect on both spin and flight qualities. They are much lighter than the other parts (*e.g.,* a joint weighs around 0.5 g while the other parts typically weigh around 5 g) and also located along the stick.

4. Simulation of Bamboo-Copter Flight

Our system simulates flight trajectories of bamboo-copters using a data-driven aerodynamics. This is basically a rigid body simulation framework, but the aerodynamic forces on wings are considered based on the *wing theory* [AD59]. Our simulation framework is based on Umetani et al.'s work [UKSI14]; however, we introduce several extensions such as a different wing discretization method.

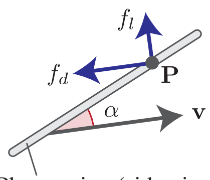
4.1. Fundamentals of Aerodynamic Force

Suppose that there is a planar wing (the wing theory is not limited to planar wings, but we consider a planar wing as an example) that moves towards its front direction at a constant velocity $\mathbf{v} \in \mathbb{R}^3$ in still air. The aerodynamic forces on this wing can be decomposed into *lift*, and *drag forces*. Lift force is the force that is perpendicular to the velocity. Drag force is the resistance force against airflow. In addition to these forces, torques are also generated; but here, we do not consider torques for simplicity.

In this paper, aerodynamic lift and drag forces on wings are estimated without running fluid simulation. Instead, we compute them based on the wing theory [AD59], which is a data-driven approach and enables much more efficient simulation. In the wing theory, the amounts of the lift force f_l and the drag force f_d are calculated as

$$(f_l, f_d)^{\mathrm{T}} = \frac{1}{2}(C_l, C_d)^{\mathrm{T}} \rho v^2 A, \quad (1)$$

where C_l and C_d are called the lift and drag coefficients respectively, and $v = \|\mathbf{v}\|$ is the air velocity relative to a wing, A is the area of the wing, and ρ is the density of air. The lift and drag forces are perpendicular and parallel to the air velocity relative to the wing (see the inset figure). These forces are applied to the *center of pressure* point $\mathbf{P}$ of the wing. Determining this point is not easy in general cases, but it is known that this theoretically locates at the quarter-chord point for symmetric airfoils without camber [AJ10], as this fact was used in the previous work [UKSI14]. The lift and drag coefficients are dimensionless and usually parametrized with the geometric form of the wing, the *angle of attack* α, and the Reynolds number Re. The angle of attack is the angle formed by the wing's front direction and the velocity. The Reynolds number is defined as $\mathrm{Re} = \rho v L / \mu$, where L is the chord length of the wing, and μ is the viscosity of the fluid. For typical geometric forms such as a taper wing, the lift and drag coefficients that are empirically obtained using large-scale fluid simulation or wind tunnels, are widely available. We refer readers to [UKSI14, AD59] for further details.



Planar wing (side view)

4.2. Settings and Notations

We deal with a bamboo-copter consisting of a stick and a few wings. In this paper, we assume that wings are planar; that is, a wing can be described by its 2-dimensional shape and its thickness. Suppose that there is a coordinate system (X, Y, Z) whose origin is at the bottom tip of the stick and whose Y direction is along the stick axis (see Figure 7). For the j-th wing, we define the mounting direction Φ^j as the angle formed by the X axis and the plane that passes through the stick axis and the wing's mounting point M. We define the mounting angle Θ^j as the angle formed by the XZ plane and the wing plane. In the current implementation, all the wings share the same attachment height h, just for simplicity. The length of the stick is represented as l.

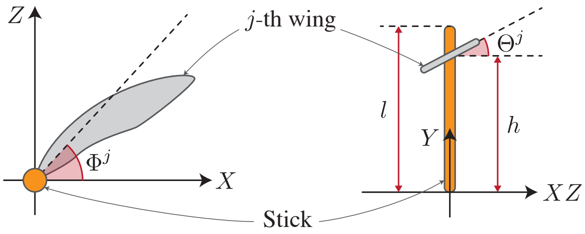
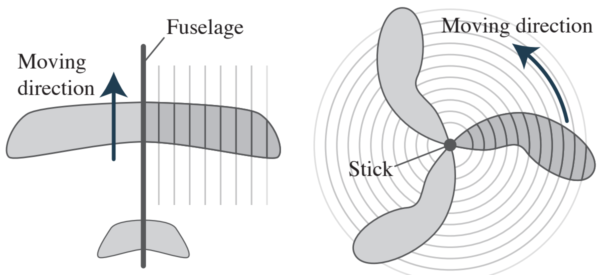


Figure 7: *Notations used in this paper. (Left) Top view of the j-th wing. (Right) Side view of the j-th wing.*



(a) Linear discretization (for model airplanes)

(b) Annular discretization (for bamboo-copters)

Figure 8: *(a) Wing element discretization for model airplanes, used in [UKSI14]. (b) Wing element discretization for bamboo-copters. We employ an* annular *wing element discretization since the dominant moving direction of bamboo-copters is rotational.*

4.3. Annular Wing Discretization

To extend the traditional wing theory for free-formed wings of model airplanes, Umetani et al. [UKSI14] introduced a method for discretizing a wing into *wing elements*, as shown in Figure 8 (a). Aerodynamic forces are computed for each wing element, and then integrated for obtaining the entire forces on the wing. We take a similar approach to deal with free-formed wings of bamboo-copters. Different from the previous work, where a wing is divided linearly, a wing is divided *annularly*, as shown in Figure 8 (b). In particular, each wing is divided into (usually around ten) wing elements by the concentric circles whose centers are the stick of the bamboo-copter. The radii of the concentric circles increase at an equal interval. This annular discretization is more suitable than the previous way because angular (not linear) velocities are dominant in our problem setting, which breaks the assumption of the previous method that each wing element moves towards the cut direction. Our wing element discretization method can be seen as a simplified version of the *blade element momentum* (BEM) theory [Ing11], and the annular discretization is used for modeling wind turbines or propellers.

4.4. Force Estimation and Rigid Body Dynamics

We simulate a bamboo-copter as a rigid body interacting with aerodynamic forces. Slight deformation might occur, but it was not visible in our observation and we decided to ignore it in our current model. In this subsection, suppose that all the variables are described in the *body-space* coordinates, whose origin is the center of mass of the bamboo-copter, which can simplify the following formulations. We refer readers to the course note provided by Baraff [Bar01] for details of rigid body dynamics.

Let w^{ij} be the i-th wing element of the j-th wing, $\mathbf{p}^{ij} \in \mathbb{R}^3$ be the position of the center of the pressure of w^{ij}, and $\mathbf{v}^{ij} \in \mathbb{R}^3$ be the velocity of $\mathbf{p}^{ij}$. We have

$$\mathbf{v}^{ij} = \omega_{\text{center}} \times \mathbf{p}^{ij} + \mathbf{v}_{\text{center}}, \tag{2}$$

where $\omega_{\text{center}} \in \mathbb{R}^3$ is the angular velocity of the bamboo-copter around the center of mass, and $\mathbf{v}_{\text{center}} \in \mathbb{R}^3$ is the linear velocity of the center of mass of the bamboo-copter. By applying Equation 1 to this wing element, the lift and drag forces on $\mathbf{p}^{ij}$, can be described in vector form as

$$\mathbf{f}_l^{ij} = \frac{1}{2} c_l^{ij} \rho (\mathbf{v}^{ij} \times \mathbf{n}^i) \times \mathbf{v}^{ij} A^{ij}, \tag{3}$$

$$\mathbf{f}_d^{ij} = \frac{1}{2} c_d^{ij} \rho \|\mathbf{v}^{ij}\| \mathbf{v}^{ij} A^{ij}, \tag{4}$$

where A^{ij} is the area of this wing element, $\mathbf{n}^i \in \mathbb{R}^3$ is the normal vector of the i-th wing surface, and the coefficients c_d^{ij} and c_l^{ij} are determined from the angle of attack α^{ij} and the Reynolds number Re^{ij} of this wing element, which will be explained in §5. The angle of attach α^{ij} is given by $\alpha^{ij} = \tan^{-1}(\|\mathbf{v}_n^{ij}\| / \|\mathbf{v}_t^{ij}\|)$, where $\mathbf{v}_n^{ij}$ and $\mathbf{v}_t^{ij}$ are the normal and tangential components of $\mathbf{v}^{ij}$ to this wing element, respectively. By integrating the forces and torques over all the wing elements, we can estimate the total force and torque on the bamboo-copter, and obtain the equation of motion as

$$m\dot{\mathbf{v}}_{\text{center}} = \sum_j \sum_i (\mathbf{f}_l^{ij} + \mathbf{f}_d^{ij}) - m\mathbf{g}, \tag{5}$$

$$\dot{\mathbf{L}} = \sum_j \sum_i \mathbf{p}^{ij} \times (\mathbf{f}_l^{ij} + \mathbf{f}_d^{ij}), \tag{6}$$

where m is the mass of the bamboo-copter, $\mathbf{L} \in \mathbb{R}^3$ is the angular momentum of the bamboo-copter, and the dot above a variable denotes its time derivative. For integrating this differential equation over time, we use the fourth-order Runge-Kutta method.

5. Fitting Aerodynamic Parameters

Following Umetani et al. [UKSI14], we take a data-driven approach to obtain aerodynamic coefficients (*i.e.,* c_d and c_l) for simulating bamboo-copters. The basic idea is the same as Umetani et al.'s method; We first capture several flight trajectories of fabricated training bamboo-copters by using video cameras, and then solve an optimization problem to obtain the aerodynamic coefficients such that the errors between acquired trajectories and simulated trajectories are minimized. Unlike Umetani et al.'s problem setting, in bamboo-copter flight, drag forces on wings affect the angular velocity transition rather than the position transition. Thus, we extend their framework so that not only the position tracking but also the angular velocity tracking is effectively employed.

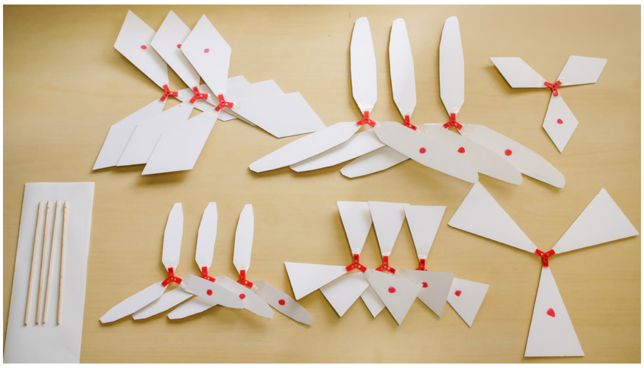


Figure 9: *Bamboo-copters used for training data acquisition.*

5.1. Aerodynamic Coefficient Representation

Function Representation as Scattered Data Interpolation. Our aerodynamic coefficient representation is mostly the same as that of Umetani et al. [UKSI14]. In our problem setting, the aerodynamic coefficient $\mathbf{c} = (c_d, c_l)$ is considered a function of the flight condition $\varepsilon = (\alpha, \mathrm{Re})$. That is, this function can be written as $\mathbf{c}(\varepsilon)$, or $\mathbf{c} : \mathbb{R}^2 \to \mathbb{R}^2 : (\alpha, \mathrm{Re}) \mapsto (c_d, c_l)$. As Umetani et al. did, we represent $\mathbf{c}(\varepsilon)$ using the radial basis function (RBF) interpolation [ALP14], which is a scattered data interpolation technique. Thus, the goal of §5 is to obtain an appropriate set of RBF data points, or *RBF centers*, from flight trajectory data. For the RBF interpolation, we use the Gaussian kernel $\phi(r) = \exp(-r^2)$.

Techniques for Effective Interpolation. Following Umetani et al., we use flat plate wings, where it is known that c_d becomes an even function and c_l becomes an odd function in terms of the angle of attack α. Thus, we can utilize the *ghost data point* method for constructing an even or odd function by RBF interpolation, which is also presented by Umetani et al.

The Reynolds numbers are usually significantly large (*e.g.,* between 2000 and 20000) compared to the angle of attack (*e.g.,* between $-\pi/2$ and $\pi/2$) in our case. Therefore, simply using the Euclidean distance of ε in the RBF interpolation results in the Reynolds numbers being too dominant. To avoid this, we use the weighted Euclidean distance, where the weights are empirically set.

Limitation. For the definition of ε, Umetani et al. considered two additional parameters that parametrize geometric forms of wing elements (thus, their ε is 4-dimensional). We could also consider these parameters in the same way, but we currently do not use them for simplicity. Unlike Umetani et al., we do not consider the pitching moment coefficient in $\mathbf{c}$, which is also a limitation.

5.2. Data Acquisition Setting

Training Bamboo-Copters. For acquiring training flight data, we prepared 14 bamboo-copters in total, shown in Figure 9, including 4 types of designs with 3 different mounting angles (*i.e.,* 15, 20, and 25 degrees), and 2 types of other designs with zero mounting angle. The same stick length is used for them all. Note that, even though

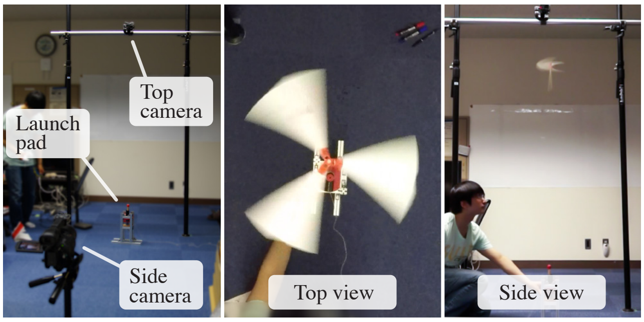
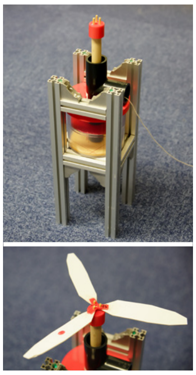


Figure 10: *The camera setting for tracking flights of bamboo-copters. The high-speed camera put on the top is used for measuring angular velocities. The camera put on the side is used for measuring the attained height of each flight.*

these bamboo-copters do not cover all the design possibilities, we could achieve a wide coverage of necessary data because both the angle of attack and the Reynolds number drastically vary during each flight. Moreover, we use a rigid body simulation as described in §4, and it can analytically predict the influence of the variable length of the stick. Thus, it is not necessary to collect data for sticks with variable length. On the other hand, rigid body simulation engine itself cannot compute the aerodynamic forces resulting from spinning, so we estimate it based on the measured data.



Launch Pad. To facilitate the launch of bamboo-copters, we constructed a launch pad as shown in the inset figure. The launch pad has a hole at its side through which a kite string passes. The stick of a bamboo-copter is put into the hole at the top, and then is rotated manually around fifteen laps so that the kite string coils up the internal pipe structure. By pulling the kite string quickly, the bamboo-copter rotates, and eventually it flies when it obtains a sufficient angular velocity. Note that we manually fly the bamboo-copters with zero mounting angle by hand, because they do not fly up direction and thus this launch pad does not work for them.

Capturing Flight Trajectories. Instead of using expensive motion capture systems, we set up a simple flight capture environment. We used two cameras, as shown in Figure 10. At the top of the launcher, we put GoPro HERO4, which can record videos at 240 FPS (frames per second). This is used for capturing angular velocities, which is why we need such a high-speed camera. At the side, we set a standard video camera, which records videos at 30 FPS. This is used for capturing attained heights during flights. We captured videos several times for each bamboo-copter. We omitted videos where bamboo-copters did not fly straight.

Acquisition of Height and Linear/Angular Velocity. From the captured videos, we manually extracted the information of heights, linear velocities, and angular velocities. We selected 3 frames for each flight video, and recorded the information at each moment. For extracting the height at a certain frame, we measured the pixel distances among the top camera, the launch pad, and the bamboo-copter in the side view. We also measured the physical distance between the top camera and the launch pad in advance. By integrating these information, we calculated the height of the bamboo-copter. For extracting the linear velocity, we checked both the right-before and right-after frames, and then read how much the bamboo-copter moves in this two-frame duration. We extracted the angular velocity in a similar way to the linear velocity, but we read how much the bamboo-copter rotates in the two-frame duration using the videos taken by the top camera.

5.3. Parameter Estimation Algorithm

The goal here is to obtain the function for computing aerodynamics coefficients $\mathbf{c} : (\alpha, \mathrm{Re}) \mapsto (c_d, c_l)$ such that simulated flights match with all the 28 recorded flights $\mathcal{F} = \{f_1, \dots, f_{28}\}$ as much as possible.

$$\arg\min_{\mathbf{c}} \sum_{f_{\mathrm{rec}} \in \mathcal{F}} C_{\mathrm{diff}}(f_{\mathrm{sim}}(\mathbf{c}), f_{\mathrm{rec}}), \tag{7}$$

where $C_{\mathrm{diff}}(\cdot,\cdot)$ represents the difference between the simulated flight f_{sim} using a given function $\mathbf{c}$ and the recorded flight f_{rec}. As we have described in the previous subsection, we measure height, linear velocity, and angular velocity of a recorded flight at three time points. We call it a flight snapshot. Suppose that a flight snapshot corresponding to the flight f_{rec} consists of three time points t_1, t_2, and t_3. For each flight snapshot, we compute the difference in the last two measurements (at t_2 and t_3) between the simulated and recorded flights. The condition in the first measurement (at t_1) is used for the initial condition of the simulation. We then add the differences for the last two measurements to obtain the difference for this specific flight. Then, C_{diff} can be written as

$$\begin{aligned} C_{\mathrm{diff}}(f_{\mathrm{sim}}(\mathbf{c}), f_{\mathrm{rec}}) = \sum_{t \in \{t_2, t_3\}} & (w_{\mathrm{height}} |h_{\mathrm{sim}}(t) - h_{\mathrm{rec}}(t)|^2 \\ & + w_{\mathrm{linear}} |v_{\mathrm{sim}}(t) - v_{\mathrm{rec}}(t)|^2 \\ & + w_{\mathrm{angular}} |\omega_{\mathrm{sim}}(t) - \omega_{\mathrm{rec}}(t)|^2), \end{aligned} \tag{8}$$

where $\{h_{\mathrm{sim}}(t), v_{\mathrm{sim}}(t), \omega_{\mathrm{sim}}(t)\}$ and $\{h_{\mathrm{rec}}(t), v_{\mathrm{rec}}(t), \omega_{\mathrm{rec}}(t)\}$ represent height, linear velocity, and angular velocity of simulated and recorded flights at the time t, respectively. We set $w_{\mathrm{height}} = 1.0, w_{\mathrm{linear}} = 0.001$, and $w_{\mathrm{angular}} = 0.00001$. Note that the value w_{angular} is much smaller than the others; however, this does not mean that we expect minimal effect from angular velocities, since the units are different.

Simulated flight for a given function $\mathbf{c}$ is computed as follows. As explained above, the initial height, linear velocity, and angular velocity are taken from the condition in the time point t_1 in the corresponding recorded flight. The system then runs rigid body flight simulation using the lift and drag forces computed using the given function $\mathbf{c}$. At each time step in the flight simulation, the system computes α and Re for each wing element, and then computes lift and drag forces for the wing element by applying the function $\mathbf{c}$ to the α and Re.

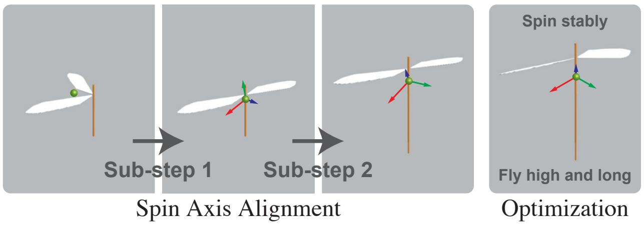


Figure 11: *Our optimization framework. First, the system considers the alignment of spin axis, which ensures the minimum requirement for stable spin. Then, the system solves the main optimization problem where both spin and flight qualities are maximized.*

The remaining question is how to represent and optimize the function **c**. Since **c** is a continuous function that maps two input values to two output values, it is difficult to optimize it directly. We therefore represent it as a summation of RBF functions and optimize the RBF centers and their coefficients. We start with a single RBF center and gradually increase the number of RBF centers in a greedy manner [BBO*09]. The first RBF center $\mathbf{k}_0$ is placed at the average of all the data samples, that is, average of α and Re of all the wing elements in all the flight snapshots. The coefficient of the RBF center $\mathbf{k}_0$ is set to a default initial value $c_d(\mathbf{k}_0) = c_l(\mathbf{k}_0) = 0$ and optimized using COBYLA [Pow98]. After optimizing $c_d(\mathbf{k}_0)$ and $c_l(\mathbf{k}_0)$, we identify a flight snapshot where the difference between the simulated and recorded flights is the largest. We pick an arbitrary wing element in the data point, and use α and Re of the wing element as the location of the next RBF kernel ($\mathbf{k}_1$). We then optimize the coefficients of $\mathbf{k}_0$ and $\mathbf{k}_1$ simultaneously using COBYLA. We repeat the above procedure multiple times. It typically takes around 1 second to run a rigid body simulation for a single flight in our current implementation. We need to run this simulation for 28 times to evaluate the cost of the current **c**. It took around 3 days to obtain 8 RBF centers.

6. Optimization of Bamboo-Copter Design

The goal of this optimization is to adjust the configuration of the bamboo-copter so that it flies well. Ideally, we would like to explicitly maximize the maximum flight height and flight duration, but it is difficult to optimize them directly. We therefore decompose the problem into two elements, spin quality and flight quality, and optimize them simultaneously. Spin quality serves a necessary condition to make the bamboo-copter stably spin around the stick. If spin quality is low, the bamboo-copter quickly looses balance and falls. Flight quality then ensures that the bamboo-copter flies higher and longer. To do so, we maximize upward force and minimize torque against spin during flight. Figure 11 describes this process.

We define spin quality as a function of principal moments of the bamboo-copter. This function assumes that an appropriate principal axis of moment is aligned with the stick of the bamboo-copter, as the minimum requirement. Thus, we adjust the bamboo-copter configuration so that an appropriate principal axis of the bamboo-copter aligns with the stick (§6.1) before running optimization of spin and flight qualities (§6.2).

6.1. Spin Axis Alignment

We run this step once before running optimization described in the next subsection. The system runs the following two sub-steps sequentially in this order.

6.1.1. Translational Alignment

In this sub-step, our system performs an optimization such that the center of mass of the bamboo-copter aligns with the stick of the bamboo-copter by changing the mounting directions $\{\Phi^j\}$ and the scales $\{s^j\}$ of the wings. To preserve the user's original intent, we define a cost function C_{preserve}:

$$C_{\text{preserve}} = \sum_j (1 - s^j)^2. \tag{9}$$

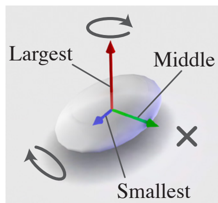
Currently we optimize uniform scaling of the wings for simplicity, and it is a future work to allow more freely modify wing shapes by using free-form deformation techniques. In this sub-step, we solve

$$\min_{\{\Phi^j\},\{s^j\}} \left\{ w_{\text{center}} C_{\text{center}} + w_{\text{preserve}} C_{\text{preserve}} \right\}, \tag{10}$$

where C_{center} is the distance from the position of the center of mass of the bamboo-copter to the stick axis in meters. In our system, we set $w_{\text{center}} = 1000$ and $w_{\text{preserve}} = 1$. We solve this minimization using COBYLA [Pow98].

6.1.2. Angular Alignment

In this sub-step, our system searches a configuration such that an appropriate principal axis of the moment of inertia aligns with the stick of the bamboo-copters. It is known that, for stable spin, the spin axis needs to be aligned to the axes of the largest or smallest principal moments of inertia [BWBSH14, HG01]. Thus, the stick needs to be aligned with the principal axis corresponds to the largest or smallest principal moment. This is a requirement to have the bamboo-copter spin stably around the stick. The optimization process described in the next section further improves the spin quality together with flight quality, but this sub-step enforces to satisfy this minimum requirement as an initialization of the subsequent optimization.



Provided that the moment of inertia tensor $\mathbf{I} \in \mathbb{R}^{3\times3}$ is a real symmetric matrix, there exists a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ that satisfies $\mathbf{RIR}^{\mathrm{T}} = \mathrm{diag}(I_a, I_b, I_c)$, where $I_a \geq I_b \geq I_c$. I_a, I_b and I_c, are called the principal moments of inertia, which are the eigenvalues of I, and the columns of $\mathbf{R}$ are called the principal axes, which are the eigenvectors corresponding to the eigenvalues I_a, I_b and I_c. Now, our goal is to adjust the bamboo-copter configuration so that the stick orientation aligns with either I_a or I_c. Specifically, the following procedure is computed. We first check the principal axis that is closest to the stick axis. If it is corresponding to the largest or smallest principal moment, then there is nothing to be done. If not, the system searches for a configuration that escapes from this situation. Starting with the initial stick length and initial mounting point given in the user's design, our system searches for the stick length l and mounting point h on the stick satisfying the requirement by breadth first search.

6.2. Optimizing Spin and Flight Quality

At this point, a principal axis of the moment of inertia is already aligned with the stick. This step further optimizes spin and flight qualities simultaneously. Specifically, we adjust the mounting directions and angles of the wings to minimize the cost function:

$$C_{\text{total}} = w_{\text{spin}} C_{\text{spin}} + w_{\text{flight}} C_{\text{flight}} + w'_{\text{preserve}} C'_{\text{preserve}}, \tag{11}$$

where C_{spin} and C_{flight} represent spin and flight qualities, respectively, and C'_{preserve} represents a cost function that encourages to preserve the original design. We set $w_{\text{spin}} = 1, w_{\text{flight}} = 1$, and $w'_{\text{preserve}} = 0.4$. We describe the details of the spin quality (§6.2.1) and flight quality (§6.2.2) in the following subsections. In this step, we define $C'_{\text{preserve}} = (1 - l/l_{\text{orig}})^2$ so that the stick length l will not drastically change from the original length l_{orig}.

We solve this optimization using gradient descent method. Although gradient descent does not guarantee that the solution is globally optimum, we adopt this because it often finds a solution close to the initial configuration, which is desirable as the initial configuration reflects the user's original design intention. The derivative of C_{total} is computed using the numerical differentiation.

The optimization adjusts the stick length l, the mounting direction Θ of each wing, and the mounting angle Φ of each wing. In contrast to the pre-process in §6.1.1, this optimization does not change scales of the wings because it can easily break the spin axis alignment, which is already satisfied by the previous step. The vertical position of the wings h is not considered as the variable of this optimization but is changed according to the stick length l such that the ratio of h to l is constant. This is also for conserving the original design intention.

6.2.1. Spin Quality

The principal axis is already aligned with the spin axis (stick direction) in the previous step, but it is not sufficient for stable spin. It is important that the difference between the principal moments is large to achieve a stable spin. We try to maximize the difference in the spin quality optimization.

Our definition of spin quality is an extension of spin quality proposed by Bächer et al. [BWBSH14]. The difference is that we additionally consider a spin around a principal axis of moment corresponding to the smallest principal moment while they only consider the largest principal moment. They considered the largest one only because spin around the axis corresponding to the largest principal moment is preferable to the smallest one in terms of kinetic energy. However, in our case, spin around an axis corresponding the smallest principal moment is common as seen in standard traditional bamboo-copters consisting of two wings. In such a case, the largest principal axis is perpendicular to the orientation of the stick and it cannot be the spin axis.

Bächer et al. [BWBSH14] defined the spin quality as

$$f_{\text{large}} = \left(\frac{I_b}{I_a}\right)^2 + \left(\frac{I_c}{I_a}\right)^2, \tag{12}$$

where I_a, I_b and I_c are the principal moments of inertia and I_a is the largest principal moment of inertia. We use this as a spin quality when the spin axis is aligned with the principal axis of moment corresponding to the largest principal moment I_a.

In addition to this, we newly define the following quality:

$$f_{\text{small}} = \left(\frac{I_c}{I_a}\right)^2 + \left(\frac{I_c}{I_b}\right)^2, \tag{13}$$

We use this when the spin axis is aligned with the principal axis of moment corresponding to the smallest principal moment I_c.

We also consider the alignment of principal axis of moment in the optimization when computing the spin quality. Specifically, we define our spin quality as follows.

$$C_{\text{spin}} = w_{\text{center}} C_{\text{center}} + w_{\text{angle}} C_{\text{angle}} + w_{\text{stability}} f_{\text{large}} \text{ (or } f_{\text{small}}) \tag{14}$$

where C_{center} is the distance between the center of mass and the stick (the same quantity used in §6.1.1) and C_{angle} is the angle between the stick and the principal axis corresponds to the largest or smallest principal moment. We set $w_{\text{center}} = 1000$, $w_{\text{angle}} = 0.1$ and $w_{\text{stability}} = 1.0$.

6.2.2. Flight Quality

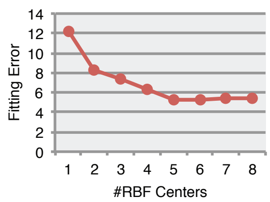
Optimization of spin quality makes the bamboo-copter spins stably, but it is not sufficient to make the bamboo-copter flies well. We want the bamboo-copter to fly high and long. To achieve this, we maximize force for upward direction and minimize drag torque against spin during the flight. Specifically, we define the cost function for improving the flight quality as

$$C_{\text{flight}} = \sum_{s \in \mathcal{S}} \{-w_{\text{up}} F_{\text{up}}^2(s) + w_{\text{drag}} T_{\text{drag}}^2(s)\}, \tag{15}$$

where $\mathcal{S}$ represents a set of linear and angular velocity samples, and $F_{\text{up}}(s)$ and $T_{\text{drag}}(s)$ represent the total upward force and the total drag torque around the stick for a pair of linear and angular velocities $s = \{\mathbf{v}, \omega\}$, respectively. We use the set of $28 \times 3 = 84$ snapshot points observed in the training data gathering (§5.2) as the definition of $\mathcal{S}$. We set $w_{\text{up}} = 5.3$ and $w_{\text{drag}} = 6000$.

7. Results

7.1. Evaluation of the Aerodynamic Parameter Fitting



Convergence. The aerodynamic coefficients of our wing elements are learned from 28 empirical flight trajectory data using 14 bamboo-copters, as described in §5.3. The inset figure shows the convergence behavior of the fitting error with the number of the RBF data points, or the RBF centers. In the current system, we use the coefficients consisting of 8 RBF centers.

Accuracy. Figure 12 shows comparisons between acquired trajectories and fitted trajectories after the fitting computation. Here, we show both the transition of heights and angular velocities. To validate the learned aerodynamic coefficients, we compared simulated flight trajectories and actual flight trajectories of bamboo-copters that are not included in the training data. Figure 13 shows the results. Although the set of training bamboo-copters does not have

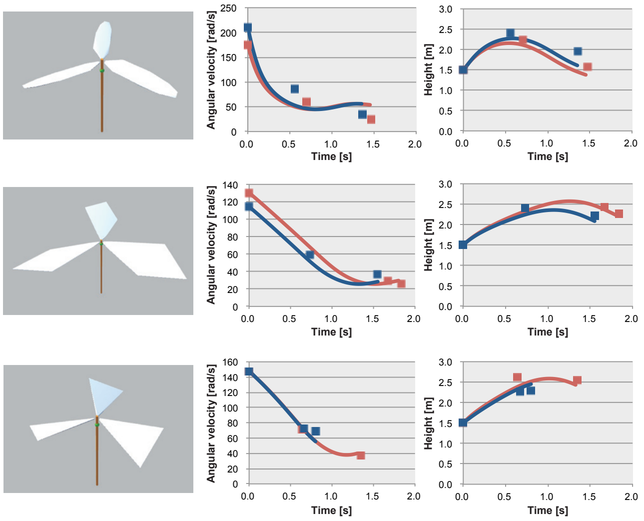


Figure 12: *Fitted trajectories. Lines represent fitted simulated trajectories, and square dots represent acquired information from recorded videos. For each bamboo-copter, we show two sets of them (colored differently). The middle column shows the angular velocity transitions* (rad/s) *and the right column shows the height transitions* (m)*. Horizontal axis in each graph indicates the elapsed time from the first acquisition time point.*

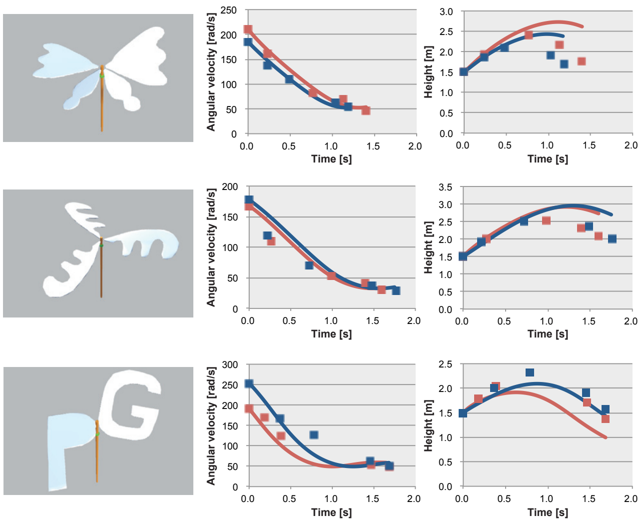


Figure 13: *Accuracy of the simulation for the bamboo-copters that are not included in the training data.*

asymmetric or 2-wing bamboo-copters, the simulated trajectories could predict the actual flight behavior with reasonable accuracy.

7.2. Evaluation of the Optimization

Our accompanying video contains a full sequence of designing a bamboo-copter, in which the optimization function is effectively used. In that sequence, a bamboo-copter that was originally very unstable becomes stably flyable over three seconds after optimization. Please see the accompanying video.

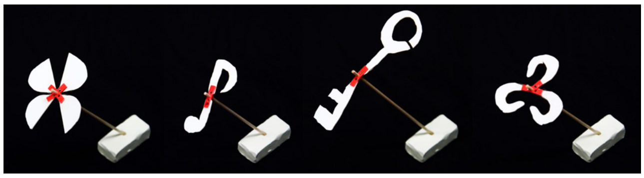


Figure 14: *Bamboo-copters designed in the informal user study.*

7.3. User Experience

Figure 1 shows the free-formed creative bamboo-copters that we designed using our system. All of them are quite different from traditional bamboo-copter design. In addition, we conducted an informal user study where four students were participated. Figure 14 shows the bamboo-copters that were designed in this study. Everyone enjoyed designing bamboo-copters using our system. Each design took around ten minutes. Please see the accompanying video for the flying animation.

8. Limitations and Future Work

User Evaluation. As novice users could design flyable creative bamboo-copters in the informal study, we believe that everyone, even children, can enjoy designing original bamboo-copters by using our system. It would be useful to conduct formal user studies and workshops to further evaluate and improve our system.

Aerodynamics Model. In flight simulation, we assume that there is no interference between wings, as in the wing theory [AD59]. Thus, if a user puts wings very closely, this would make our simulation less accurate. We did not consider the pitching moment on each wing element in our flight simulation. This is not a problem when the bamboo-copter is symmetric, as pitching moments will be canceled out. However, in the case of asymmetric bamboo-copters, as pitching moments might affect the simulation results. In our experiments, we have not observed any case where the behavior is noticeably different between simulation and actual flight. Our current parametrization of wing element is simple and considers only angle of attack and Reynolds number; incorporating other characteristics could improve the simulation quality, which is a future work. Our model assumes that rotational movement is dominant; thus, once a bamboo-copter loses most of the angular momentum by drag force, the simulation is no longer valid.

Fitting Accuracy. We manually designed 14 bamboo-copters for gathering training flight data. As discussed, even though the number of training bamboo-copters is not large, we could achieve a wide coverage of necessary data because both the angle of attack and the Reynolds number vary drastically during each flight. Nevertheless, gathering more flight data could improve the accuracy of flight prediction. Also, as a future work, it is desirable to more systematically generate training bamboo-copters to effectively learn the aerodynamic parameters. We consider that our current data acquisition setting is not very accurate. To obtain more accurate data, a possible approach is to use a professional motion capture system.

Optimization. Our cost function to evaluate spin and flight qualities does not consider the effect of aerodynamic force balance on spin stability. We defined the cost function f_{small} to evaluate spin quality in §6.2.1. Though we have observed that it works as expected, we consider that validations of f_{small} (and also f_{large} defined by [BWBSH14]) are necessary in the future.

Beyond Planar Wing. It is an important future work to consider 3-dimensional wing shapes, beyond planar wings. This will provide two benefits. First, planar wings are known to generate much larger drag forces than streamlined wings, and thus using 3-dimensional wings could enable longer flight. Second, it expands the design space, and enables users to design more creative bamboo-copters.

9. Conclusion

We presented an interactive design system for free-formed bamboo-copters. This system can visualize a simulated flight trajectory during the user's edit, and provides an opportunity of using automatic design optimization. To enable this system, we presented a data-driven aerodynamic simulation framework, which is an extension of Umetani et al.'s work [UKSI14], and formulated how a bamboo-copter can be optimized such that it spins stably and flies high and long. We fabricated various creative bamboo-copters designed with our system, including highly asymmetric ones.

Acknowledgments We would like to thank Nobuyuki Umetani and Haoran Xie for insightful discussions and anonymous reviewers for their valuable comments. Yuki Koyama is funded by JSPS research fellowship. This work was supported by JSPS KAKENHI Grant Number 26240027 and 26-8574.

References

[AD59] Abbott I. H. A., Doenhoff A. E. v.: *Theory of wing sections: including a summary of airfoil data*, corr. ed. Dover Publications, 1959. 3, 4, 9

[AJ10] Anderson Jr J. D.: *Fundamentals of aerodynamics*, 5th ed. McGraw-Hill Education, 2010. 4

[ALP14] Anjyo K., Lewis J. P., Pighin F.: Scattered data interpolation for computer graphics. In *SIGGRAPH 2014 Courses* (2014), pp. 27:1–27:69. 5

[Bar01] Baraff D.: Physically based modeling: Rigid body simulation. In *SIGGRAPH 2001 Course Notes* (2001). URL: http://www.pixar.com/companyinfo/research/pbm2001/pdf/notesg.pdf. 5

[BBO*09] Bickel B., Bächer M., Otaduy M. A., Matusik W., Pfister H., Gross M.: Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. 28*, 3 (July 2009), 89:1–89:9. 2, 7

[BWBSH14] Bächer M., Whiting E., Bickel B., Sorkine-Hornung O.: Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph. 33*, 4 (July 2014), 96:1–96:10. 2, 7, 8, 10

[HG01] Herbert Goldstein Charles Poole J. S.: *Classical mechanics*, 3rd ed. Addison Wesley, 2001. 7

[Ing11] Ingram G.:. Wind turbine blade analysis using the blade element momentum method (version 1.1) [online]. 2011. URL: https://community.dur.ac.uk/g.l.ingram/download/wind_turbine_design.pdf [cited 5/24/2016]. 4

[JWL*13] Ju E., Won J., Lee J., Choi B., Noh J., Choi M. G.: Data-driven control of flapping flight. *ACM Trans. Graph. 32*, 5 (Oct. 2013), 151:1–151:12. 2

[KSS*15] Koyama Y., Sueda S., Steinhardt E., Igarashi T., Shamir A., Matusik W.: Autoconnect: Computational design of 3d-printable connectors. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 231:1–231:11. 2

[Lei06] Leishman J. G.: *Principles of Helicopter Aerodynamics with CD Extra*, 2nd ed. Cambridge university press, 2006. 1

[LSZ*14] Lu L., Sharf A., Zhao H., Wei Y., Fan Q., Chen X., Savoye Y., Tu C., Cohen-Or D., Chen B.: Build-to-last: Strength to weight 3d printed objects. *ACM Trans. Graph. 33*, 4 (July 2014), 97:1–97:10. 2

[MBT*12] Miguel E., Bradley D., Thomaszewski B., Bickel B., Matusik W., Otaduy M. A., Marschner S.: Data-driven estimation of cloth simulation models. *Comput. Graph. Forum 31*, 2pt2 (May 2012), 519–528. 2

[MUB15] Martin T., Umetani N., Bickel B.: Omniad: Data-driven omni-directional aerodynamics. *ACM Trans. Graph. 34*, 4 (July 2015), 113:1–113:12. 2

[PDJ*01] Pai D. K., Doel K. v. d., James D. L., Lang J., Lloyd J. E., Richmond J. L., Yau S. H.: Scanning physical interaction behavior of 3d objects. In *Proc. SIGGRAPH '01* (2001), pp. 87–96. 2

[Pow98] Powell M. J. D.: Direct search algorithms for optimization calculations. *Acta Numerica 7* (1 1998), 287–336. 7

[PWLSH13] Prévost R., Whiting E., Lefebvre S., Sorkine-Hornung O.: Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph. 32*, 4 (July 2013), 81:1–81:10. 2

[PZM*15] Panetta J., Zhou Q., Malomo L., Pietroni N., Cignoni P., Zorin D.: Elastic textures for additive fabrication. *ACM Trans. Graph. 34*, 4 (July 2015), 135:1–135:12. 2

[SBR*15] Schumacher C., Bickel B., Rys J., Marschner S., Daraio C., Gross M.: Microstructures to control elasticity in 3d printing. *ACM Trans. Graph. 34*, 4 (July 2015), 136:1–136:13. 2

[SLMI11] Saul G., Lau M., Mitani J., Igarashi T.: Sketchchair: An all-in-one chair design system for end users. In *Proc. TEI '11* (2011), pp. 73–80. 2

[SVB*12] Stava O., Vanek J., Benes B., Carr N., Měch R.: Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph. 31*, 4 (July 2012), 48:1–48:11. 2

[UIM12] Umetani N., Igarashi T., Mitra N. J.: Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph. 31*, 4 (July 2012), 86:1–86:11. 2

[UKSI14] Umetani N., Koyama Y., Schmidt R., Igarashi T.: Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. 33*, 4 (July 2014), 65:1–65:10. 1, 2, 4, 5, 10

[UTMI11] Umetani N., Takayama K., Mitani J., Igarashi T.: A responsive finite element method to aid interactive geometric modeling. *IEEE Comput. Graph. Appl. 31*, 5 (Sept. 2011), 43–53. 2

[WP03] Wu J.-c., Popović Z.: Realistic modeling of bird flight animations. *ACM Trans. Graph. 22*, 3 (July 2003), 888–895. 2

[WW16] Wang L., Whiting E.: Buoyancy optimization for computational fabrication. *Comput. Graph. Forum 35*, 2 (2016), 49–58. 2