OmniAD: Data-driven Omni-directional Aerodynamics

Tobias Martin* ETH Zürich
Nobuyuki Umetani* Disney Research Zürich
Bernd Bickel IST Austria

(a) example falling motion (b) omni-directional aerodynamics model (c) real-time simulation (d) flyable kite design

Figure 1: *(a) Example sequence of a falling heart-shaped foam object. (b) Acquired omni-directional aerodynamic property for force and torque generated by air. (c) Real-time simulation of many hearts. (d) Flyable design of a kite carrying the heart inside.*

Abstract

This paper introduces "OmniAD," a novel data-driven pipeline to model and acquire the aerodynamics of three-dimensional rigid objects. Traditionally, aerodynamics are examined through elaborate wind tunnel experiments or expensive fluid dynamics computations, and are only measured for a small number of discrete wind directions. OmniAD allows the evaluation of aerodynamic forces, such as drag and lift, for any incoming wind direction using a novel representation based on spherical harmonics. Our data-driven technique acquires the aerodynamic properties of an object simply by capturing its falling motion using a single camera. Once model parameters are estimated, OmniAD enables realistic real-time simulation of rigid bodies, such as the tumbling and gliding of leaves, without simulating the surrounding air. In addition, we propose an intuitive user interface based on OmniAD to interactively design three-dimensional kites that actually fly. Various non-traditional kites were designed to demonstrate the physical validity of our model.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: aerodynamics, real-time animation, data-driven physics, fabrication, kites

*Tobias Martin and Nobuyuki Umetani are joint first authors

1 Introduction

The interaction between lightweight objects and air results in intricate motions and stunning aerodynamic effects. For example, leaves falling to the ground, while flipping, tumbling, and gliding through the air, are a fascinating natural phenomenon. Ingenious inventions, such as kites, sailboats, windmills, and airplanes, harness the power of aerodynamics and have brought tremendous benefits to mankind. The resulting motion of these objects is determined partially by the interplay between the object's surface geometry and the dynamics of the surrounding air. There are numerous applications in science and engineering that rely on simulating this effect, from animating a simple autumn scene in a computer game to the computational design of complex aircraft.

Traditionally, wind tunnels and computational fluid dynamics (CFD) simulations have been used to measure and predict aerodynamic forces. Such elaborate machinery, however, usually requires time-consuming set-up, is computationally expensive, and is only accessible to small numbers of domain experts. This limits the potential applications in animation and in computational design tools for a more general audience.

In this paper, we present *Omni-directional AeroDynamics (OmniAD)*, a novel data-driven representation and modeling technique for simulating the aerodynamic effects of lightweight objects. Our method is tailored to allow the simple acquisition of aerodynamic properties, requiring only a single camera, and is targeted for interactive application scenarios, carefully balancing simulation speed and accuracy. Our model operates in real-time, adds only a negligible overhead to the computational cost, and can be integrated into existing rigid-body simulators. For example, the complex motion of falling objects, such as hundreds of leaves, can be animated with our model in real-time. Moreover, the quantitative accuracy of our model allows the interactive design of real-world aerodynamically functional objects.

OmniAD is data-driven and all required model data are acquired by recording videos of falling objects with a single camera. From these input videos, the three-dimensional (3D) motion trajectory is recon-

structed and used to estimate the parameters of the model to fit the force and torque reaction with respect to the motion. Through this simple set-up, ordinary users can obtain the aerodynamic properties of 3D objects, without access to wind tunnels or the computational power needed for CFD simulations.

Earlier work in computer graphics proposed data-driven modeling of aerodynamic effects. Umetani et al. [2014] presented an interactive glider design system. Their model was based on wing theory [Abbott 1959], a widely used model in engineering, to estimate drag and lift forces acting on airfoils. However, wing theory is typically limited to flat and symmetrical two-dimensional (2D) shapes, where the airflow direction is included in the plane of symmetry. In contrast, our aerodynamics model targets general 3D objects, which is necessary for 3D free-form designs. Such a model is significantly more challenging than a 2D symmetrical model, because establishing the parameterization of the 3D aerodynamic forces and torques that vary with the object's 3D orientation is much more complex.

To validate our method, we demonstrate the capabilities of OmniAD in the context of interactive hobby-grade flyable 3D kite design. Kites stay in the air with a subtle balance of aerodynamic lift force, drag force, and torques. Thus, original free-form kite design is very challenging. Our system allows the user to interactively explore kite design by displaying equilibria and the stability of the aerodynamics forces in real-time. A user can intuitively explore flyable free-form kites by arranging 3D primitive shapes whose aerodynamics are studied beforehand with our data-driven technique. The system outputs 3D printable parts for the rapid fabrication of original box kite designs. We demonstrate our design interface with the design of four flyable non-traditional 3D kites.

This paper makes three main contributions:

- We formulate an aerodynamic model that handles omni-directional airflow.
- We present a data-driven framework that features lightweight acquisition of aerodynamic properties.
- We demonstrate an interactive design system for flyable free-form 3D kites.

2 Related Work

Aerodynamic model. In the field of aerodynamics, forces and moments resulting from the motion of air acting on an object have been studied extensively for the design of aerodynamically functional objects, such as aircraft. For example, the aerodynamic properties of airfoils are well known [Abbott 1959]. While most of the aerodynamics models assume steady flow, unsteady aerodynamics are modeled for specific objects such as falling 2D plates [Andersen et al. 2005], falling disks [Zhong et al. 2011], and 2D flapping wings [Wang et al. 2004].

In computer graphics, the aerodynamic behavior of wings has been studied for animating [Wu and Popović 2003] and controlling [Ju et al. 2013] flapping birds. It has also been used for simulating the flapping of cloth based on drag and lift resulting from a wind field [Stam 2009]. However, these models typically assume a limited range of wing configurations (roughly $\pm 20^\circ$ in angle of attack), because such flying objects are designed to move in a single direction. Except for airfoils and a small class of primitive shapes, aerodynamics forces are unknown and have to be determined through expensive CFD simulations or time-consuming wind tunnel experiments.

For graphical applications, various aerodynamics models have been presented for real-time animation of 3D objects in the air. Wejchert et al. [1991] presented an aerodynamic force, given a triangle mesh, summing up drag and traction forces for each triangle independently. However, the model ignores lift force and torque, and is not directly applicable to objects that stay airborne with lift force. Two studies [Yuan et al. 2011; Xie and Miyata 2013] achieved chaotic aerodynamics motions by applying a stochastic force to the object. However, these studies lack estimations of aerodynamics forces, which depend on the shape and orientation of a given object. Recently, Weißmann et al. [2012] reported that the wobbling behavior of rigid objects falling within a dense fluid (e.g., water) can be simulated by adding anisotropic mass to the object. If the fluid's density is significantly less than the object's density, this *add-mass effect* can be ignored [Zhong et al. 2011]. Furthermore, the model is based on the potential flow theory, which does not provide any aerodynamics forces under a steady flow (*d'Alembert paradox*) [Batchelor 2000] and thus is not applicable for simulating kites that fly under steady flow.

Rigid body-fluid interaction in graphics. This paper focuses on the one-way interaction between fluid and rigid bodies for interactive design applications. Our method is able to animate thousands of fully immersed rigid bodies in real-time, ignoring pair-wise coupling between the rigid bodies. Requiring much more computation time, solving for two-way interactions is an extensively studied subject in computer graphics. Carlson et al. [2004] simulate coupling using a grid-based fluid simulation and treating the rigid objects as if they were made of fluid. The rigidity is maintained by constraining the velocities of the region covered by an object to be rigid body motion. Klingner et al. [2006], Batty et al. [2007] and Robinson-Mosher et al. [2008] consider implicit two-way coupling of fluids to rigid bodies, and Chentanez et al. [2006] propose a method for coupling of fluids and deformable solids. Interactions between SPH particles and rigid bodies also have been studied for open surface fluids [Ihmsen et al. 2014]. Although these methods produce high-fidelity results, only a few of them are suitable for real-time applications. Modal analysis has been used for real-time fluid-solid interactions [Treuille et al. 2006]; however, the method requires expensive pre-computations for a predefined fluid domain. Our focus is on interactive design of aerodynamically functional objects and we restrict ourselves to scenarios where one-way coupling provides sufficient accuracy. For example, our kite design system prevents close placement of two objects to avoid interference, which could only be taken fully into account with more expensive two-way coupling simulations. In practice, we observed that even with such a design restriction, we can span a highly interesting design space for functional aerodynamic objects.

Measurement-based parameter fitting. Our aerodynamic model is formulated in such a way that the behavior of the simulated object agrees with the behavior of its real-world counterpart by fitting model parameters to measured data. Data-driven approaches have been used for a wide range of physical phenomena. The seminal work of Pai et al. [2001] modeled the deformation behavior of objects based on captured data. Bickel et al. [2009] proposed an image-based approach that acquires the heterogeneous and anisotropic deformation of a given solid. Physical parameters of cloth deformation have also been estimated in several studies [Wang et al. 2011; Miguel et al. 2012; Miguel et al. 2013].

Our work was inspired by *Pteromys* [Umetani et al. 2014], a system that captures the parameters of a wing model for interactively designing arbitrarily shaped paper airplanes. However, in contrast to the acquisition system in *Pteromys*, which assumes symmetric 2D shapes flying in the plane of symmetry, our method is able to capture and simulate the 3D omni-directional aerodynamic properties of a given 3D object.

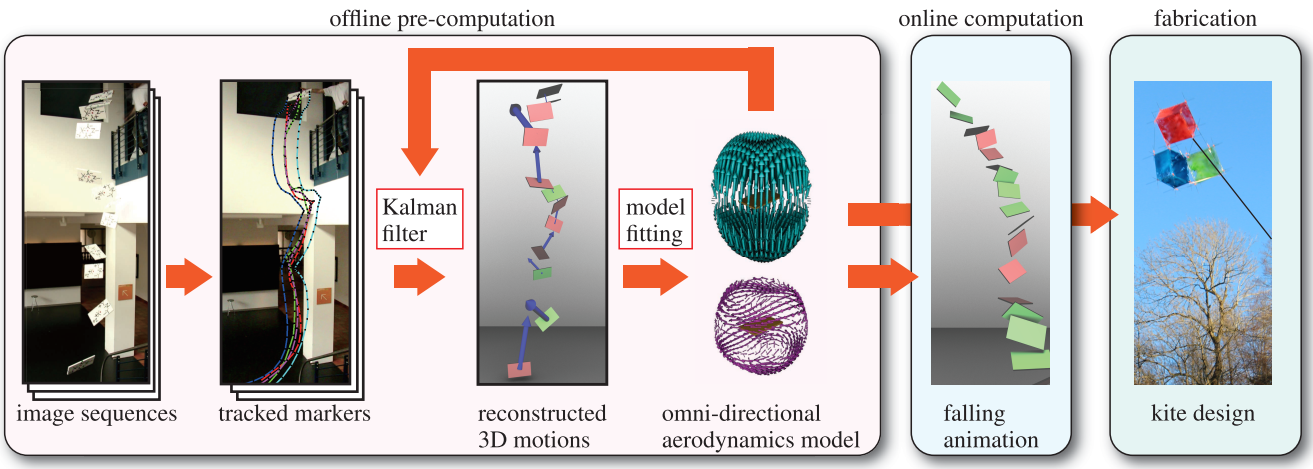


Figure 2: *An overview of our system. An object with attached markers is recorded while being dropped. From the image data, the 3D motion trajectory and omni-directional aerodynamic properties are determined. Our data-driven model can be used to simulate the aerodynamic behavior of objects in real-time and supports interactive design of aerodynamically functional objects such as kites.*

Fabrication-oriented design. Computer graphics have already made significant contributions towards systems that facilitate interactive and intuitive design of real-world objects. Motivated by advanced computer-controlled fabrication techniques, researchers have started investigating the design and reproduction of various properties, such as appearance [Hullin et al. 2013], approximation of shape [McCrae et al. 2011; Hildebrand et al. 2012; Schwartzburg and Pauly 2013; Cignoni et al. 2014], deformation behavior [Bickel et al. 2010], and a general combination of these [Chen et al. 2013]. In addition, several approaches have been presented for optimizing functional properties: for example, maximizing structural strength [Stava et al. 2012; Lu et al. 2014] and mechanical structures [Ceylan et al. 2013; Zhu et al. 2012].

At a conceptual level, we share similar goals to the computational approaches that make an object stand by redistributing its mass [Prévost et al. 2013], or ensuring rotational stability [Bächer et al. 2014]. While these studies optimize orientation stability under gravitational or inertial force that is given by an analytical formula, we target stability under much more complex aerodynamic forces.

Inspired by the seminal work of *Plushie* [Mori and Igarashi 2007], an interactive design system for plush toys, several simulation-in-the-loop systems have been proposed that ensure the designer's control over esthetic considerations while maintaining functionality. Interactivity helps to explore complex design spaces, for example for designing physically valid furniture [Saul et al. 2011; Umetani et al. 2012], frame structures [Song et al. 2013], inflatable balloons [Skouras et al. 2014], a glider [Umetani et al. 2014], mechanical automata [Coros et al. 2013], and clothing [Umetani et al. 2011]. At the core of these approaches are physics-based simulation models that carefully balance accuracy and rapid responses for either rigid-body interactions, solids, or shells. Here, we present a novel aerodynamics model for 3D objects.

3 Overview

The goal of our system is to provide a workflow that can easily acquire the aerodynamic properties of real-world objects, efficiently simulate their behavior, and make the design of lightweight objects that can fly intuitive and efficient. The foremost challenge in our system is how to represent and acquire omni-directional aerodynamics for a given rigid body. For representation, we use spherical harmonics to extend traditional aerodynamic models based on dimensional analysis to a novel omni-directional aerodynamic model, which allows handling of arbitrary airflow directions. For acquisition, we present a scheme that estimates parameters for the omni-directional aerodynamic model from video sequences of falling objects (e.g., a unit of a box kite). These two aspects in combination constitute the proposed omni-directional aerodynamics pipeline, called *OmniAD*.

Fig. 2 gives an overview of our OmniAD system. First, markers are attached to the object. Then the object is dropped multiple times with varying starting configurations, where each drop is recorded by a single video camera. From these video sequences, our system reconstructs the 3D motion of the object, based on the locations of the markers (§ 5), and infers the aerodynamic forces. Then the system estimates the parameters of the model, fitting the motion and aerodynamic forces of the object to the reconstructed data (§ 6). Based on this representation, we can simulate the physical behavior of hundreds of objects at interactive rates (§ 4.3). This interactive simulation method also forms the core of our design system for box kites, called *KiteShop*. Our design interface allows users to explore novel kite designs by combining primitive shapes with known OmniAD properties (§ 7). The system features operations such as placing, scaling, and continuously manipulating the geometry of parameterized primitives. A feasible kite design is characterized by an equilibrium of lift forces, drag forces, torque, and string tension forces. *KiteShop* visualizes these quantities in real-time, helping users to gain an intuitive understanding of their design choices. Finally, the system outputs connectors that can be printed to conveniently build real-world kites. Before introducing the proposed omni-directional model (§ 4), we first specify the equation of motion, and then describe the target setting.

Notation. In this paper, bold face letters indicate matrices (e.g., $\mathbf{R}$). Arrows above symbols indicate vectors (e.g., $\vec{v}$). Uppercase letters are used for values in the body-fixed frame, and lowercase letters are used for the corresponding values in the global frame.

Equation of motion. We assume that a falling object is a rigid body, where its configuration at time t is described by its center of gravity $\vec{y}(t)$ and its orientation by $\mathbf{R}(t) \in SO(3)$. More formally, a point $\vec{X} \in \mathbb{R}^3$ in the body-fixed frame of the rigid body moves in

the global frame as $\vec{x}(t) = \vec{y}(t) + \mathbf{R}(t)\vec{X}$. Note that the origin of the body is the body's center of mass (see Fig. 3).

We use a global frame to represent the velocity of the object's center of gravity, and use a body-fixed frame for the angular velocity representation. The velocity of the center of gravity in the global frame is written as $\vec{v} = \dot{\vec{y}}$. When the wind is blowing with velocity $\vec{v}_{wind}$, the velocity of the rigid body against the air in the body-fixed coordinate can be written as $\vec{V} = \mathbf{R}^T(\vec{v} - \vec{v}_{wind})$. The angular velocity in the body-fixed coordinate frame then becomes $[\vec{\Omega}] = \mathbf{R}^T\dot{\mathbf{R}}$, where $[\vec{a}]$ denotes a 3×3 skew matrix of vector $\vec{a}$.

The equation of motion specifies how the velocity and angular velocity change with time. From Euler's equations, the equation of motion can be written as:

$$M\dot{\vec{v}} = M\vec{g} + \mathbf{R}\vec{F}, \tag{1}$$

$$\mathbf{I}\dot{\vec{\Omega}} = -[\vec{\Omega}]\mathbf{I}\vec{\Omega} + \vec{T}, \tag{2}$$

where $M \in \mathbb{R}$ is the mass of the object, and $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the rotational inertia tensor of the object in the body-fixed frame ($\mathbf{I}$ is constant over time), $\vec{T} \in \mathbb{R}^3$ and $\vec{F} \in \mathbb{R}^3$ are the torque and linear force, respectively, induced by air in the body-fixed frame. See the supplemental material for detailed derivation of (2), which is the Euler's rotation equation in the body fixed frame.

Target setting. Our model is tailored to rigid bodies that are light enough to exhibit aerodynamic effects (e.g., flying kites). We denote L as the representative length of the object, which is defined as the diagonal length of the object's bounding box. Our example objects range from 30 cm to 1 m in L, and 50 g and 200 g in mass M. The airspeed varies from 1 m/s to 10 m/s. The Reynolds number ($\mathrm{Re} = \rho L|\vec{V}|/\mu$) indicates whether the dynamics of the surrounding air contains turbulence, where ρ is fluid density and μ is fluid viscosity. Given the properties of air (we use $\rho = 1.2$ kg/m^3 and $\mu = 1.8 \times 10^{-5}$ kg/ms throughout this paper), the large Reynolds number ($\mathrm{Re} > 10^4$) suggests the all flows are in turbulent states. Note that the fluid around an object produces turbulence immediately after it starts falling. For example, a 30 cm diameter sphere with weight of 50 g reaches $\mathrm{Re} = 2 \times 10^4$ within 0.1 s when it travels down 6 cm on its way towards the ground.

4 Omni-directional Aerodynamic Model

Our aerodynamic model is based on a dimensionality analysis, and several assumptions allow a formulation that makes it applicable to a wide variety of phenomena associated with turbulent flow. We do not intend to provide exact aerodynamics forces or torques. Instead, we develop a model that agrees with most of our assumed target situations, and which makes the following reasonable assumptions:

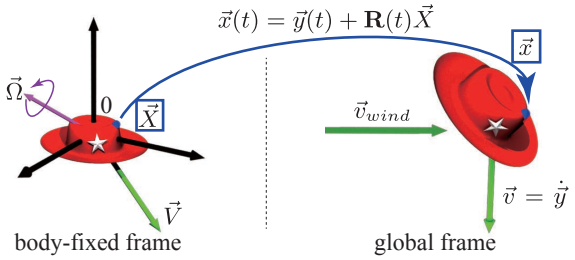


Figure 3: *Definition of the coordinate systems.*

Quasi-static assumption. We assume that the object's acceleration is low, which means that the inertia of the air can be ignored. Furthermore, if the acceleration is low, velocity changes slowly and the flow reaches a quasi-static state, which allows us to ignore the history of the object's motion. Specifically, the aerodynamic forces depend solely on the current velocity and angular velocity $\langle\vec{V}, \vec{\Omega}\rangle \rightarrow (\vec{F}, \vec{T})$. A well-known history-dependent phenomenon is fluttering, the self-induced vibration of an elastic object in flow. However, our focus is on rigid objects and thus fluttering is beyond the scope of this work.

Small angular velocity assumption. We assume that the angular velocity of the object is small compared to the linear velocity. Hence, coupling between linear velocity and angular velocity can be ignored. Specifically, the aerodynamics force and torque are given as $(\vec{F}, \vec{T}) = (\vec{F}_V + \vec{F}_\Omega, \vec{T}_V + \vec{T}_\Omega)$, where $(\vec{F}_V, \vec{T}_V)$ are the forces where the angular velocity is zero $\langle\vec{V}, 0\rangle$, and $(\vec{F}_\Omega, \vec{T}_\Omega)$ is the force and torque where the linear velocity is zero $\langle 0, \vec{\Omega}\rangle$. Ignoring the coupling between linear velocity and angular velocity means that in our model the Magnus effect [Batchelor 2000] (e.g., curved motion of a spinning ball) cannot be observed. Note that Magnus effects are visible only when the angular rotation is large, and thus are negligible for falling rigid bodies or kites.

4.1 Dimensional Analysis of Aerodynamic Forces

In this section we will introduce the background of dimensionality analysis and how we model aerodynamic forces and torque.

Buckingham π theorem. Modeling forces from complex turbulence flow is difficult because turbulence contains a large number of small vortexes. However, the model can be drastically simplified using dimensional analysis by applying the Buckingham π theorem [Batchelor 2000]. The theorem allows the derivation of the minimum number of dimensionless parameters that parameterize a physical phenomenon by analyzing the physical dimensions of the involved physical variables. This theorem is powerful because it provides a method to obtain a revised form of the involved variables in terms of non-dimensional parameters. The number of parameters can be reduced, and the fact that they are dimensionless makes them independent of the unit system used in the measurements. Furthermore, using this theorem we can derive relationships between force and involved physical quantities (e.g., wind speed, object size) without the need to know the governing equation (e.g., the Navier-Stokes equation). For an extensive discussion, we refer the reader to Batchelor [2000].

Linear velocity dependent forces. We first apply a dimensional analysis to a problem where angular velocity is zero to obtain linear velocity dependent force and torque (see supplemental material for a detailed derivation). We obtain the non-dimensional parameters Re and $\vec{\mathcal{C}}_F \in \mathbb{R}^3$ and $\vec{\mathcal{C}}_T \in \mathbb{R}^3$, allowing us to describe aerodynamic forces as:

$$\vec{F}_V = \frac{1}{2}\rho|\vec{V}|^2\vec{\mathcal{C}}_F L^2, \quad \vec{T}_V = \frac{1}{2}\rho|\vec{V}|^2\vec{\mathcal{C}}_T L^3. \tag{3}$$

The aerodynamic coefficients $\vec{\mathcal{C}}_F$ and $\vec{\mathcal{C}}_T$ depend on the object's normalized shape ($L = 1$) and its moving direction against the air $\vec{E} = \vec{V}/|\vec{V}|$. These coefficients are used widely in engineering to measure how airplanes and cars perform.

In a strict sense, these coefficients are also a function of Re. However, in the state of turbulence, the aerodynamic coefficient is insensitive to Re (*Newton's law of resistance* [Batchelor 2000]). Hence,

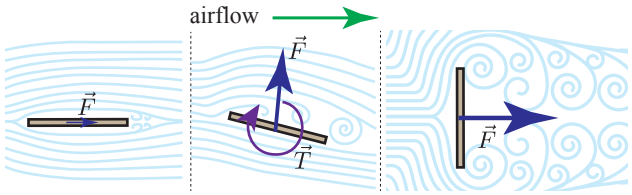


Figure 4: *Anisotropy of the aerodynamic forces. The aerodynamic force (dark blue) and torque (purple) change significantly with respect to the airflow direction relative to the object.*

the aerodynamic coefficients can be modeled as constant to Re. Note that this model captures time-averaged aerodynamic force, ignoring the fluctuation caused by vortex shedding. Such fluctuation force is typically negligible compared to the averaged force.

Angular velocity dependent forces. Next, we apply dimensional analysis to a problem where the linear velocity is zero to obtain angular velocity dependent force and torque (see supplemental material for a detailed derivation). From this we obtain

$$\vec{F}_\Omega = \frac{1}{2}\rho|\vec{\Omega}|^2\vec{\mathcal{D}}_F L^4, \quad \vec{T}_\Omega = \frac{1}{2}\rho|\vec{\Omega}|^2\vec{\mathcal{D}}_T L^5, \tag{4}$$

where $\vec{\mathcal{D}}_F \in \mathbb{R}^3$ and $\vec{\mathcal{D}}_T \in \mathbb{R}^3$ are coefficients, which depend on the object's normalized shape and normalized angular velocity $\vec{\Omega}/|\vec{\Omega}|$. By assuming a small angular velocity, the angular-velocity induced linear force $\vec{F}_\Omega = 0$ can be ignored. This means that propeller-like thrust force (e.g., flying boomerang or maple seed) are ignored. Moreover, we simplify the torque coefficient $\vec{\mathcal{D}}_T$ as $|\vec{\Omega}|^2\vec{\mathcal{D}}_T = -|\vec{\Omega}|\mathbf{D}\vec{\Omega}$, where $\mathbf{D} \in \mathbb{R}^{3\times3}$ is a positive definite symmetric matrix, called the rotational damping coefficient. This coefficient produces torque, which counteracts, i.e., decreases, angular velocity.

To summarize, aerodynamic forces and torque are modeled as

$$\vec{F} = \frac{1}{2}\rho|\vec{V}|^2\vec{\mathcal{C}}_F(\vec{E})L^2, \tag{5}$$

$$\vec{T} = \frac{1}{2}\rho\left\{|\vec{V}|^2\vec{\mathcal{C}}_T(\vec{E}) - \mathbf{D}|\vec{\Omega}|\vec{\Omega}L^2\right\}L^3. \tag{6}$$

4.2 Omni-directional Aerodynamic Coefficients

This section describes how to parameterize the proposed aerodynamic model, OmniAD. As discussed above, the aerodynamics force is modeled with $\vec{\mathcal{C}}_F(\vec{E})$ and $\vec{\mathcal{C}}_T(\vec{E})$, which are two functions that map points on the unit sphere to 3D vectors. Modeling these functions is challenging, because their values change greatly with respect to the relative airflow direction $\vec{E}$ (see Fig. 4). Due to this direction-sensitive nature, the aerodynamics force cannot generally be modeled with a linear model.

Spherical harmonics for aerodynamic coefficients. Spherical harmonics (SH) are used to represent the functions $\vec{\mathcal{C}}_F(\vec{E})$ and $\vec{\mathcal{C}}_T(\vec{E})$ defined on the unit sphere. Spherical harmonics is a set of solutions to Laplace's equation on the unit sphere. These solutions can be used as a basis to represent an arbitrary smooth scalar function, defined on the unit sphere. In practice, we observed that our data changes smoothly over the unit sphere, and therefore SH are well suited for approximating it. In the field of computer graphics, SH are used, for example, for lightweight environmental maps [Ramamoorthi and Hanrahan 2002], and for the rotation-invariant shape descriptors [Kazhdan et al. 2003].

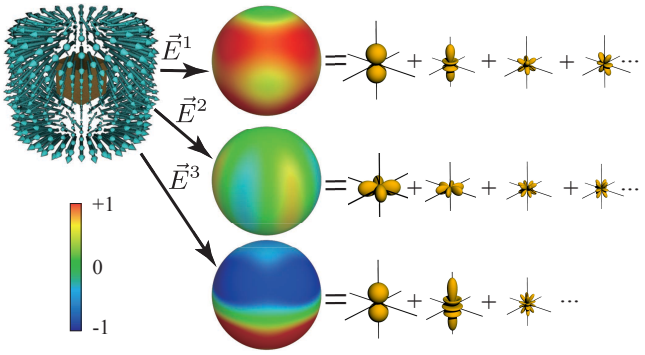
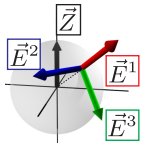


Figure 5: *Spherical harmonics representation of an aerodynamics coefficient. The cyan arrows and the color contour are an actual force coefficient value computed for a cube where two opposing sides are open. The top sphere shows variation in the drag force and middle and bottom spheres show how lift force varies with respect to the incoming air direction.*



Because the aerodynamic coefficients are 3D vector functions, we first decompose the vector coefficient with three ortho-normal bases and then represent each component of each basis with SH (see Fig. 5). Given a direction $\vec{Z}$, we establish bases as $\vec{E}^1 = \vec{E}$, $\vec{E}^2 = (\vec{E}\times\vec{Z})/|\vec{E}\times\vec{Z}|$, $\vec{E}^3 = \vec{E}\times\vec{E}^2$. These bases can be seen as radial ($\vec{E}^1$), longitudinal ($\vec{E}^2$), and latitudinal ($\vec{E}^3$) directions for a given point $\vec{E}$ on a unit sphere with a north-pole direction $\vec{Z}$. The spherical harmonics representation is insensitive to the choice of $\vec{Z}$, and here we manually choose $\vec{Z}$ in such a direction that the object is most axially symmetrical around it. This vector field representation has two singular points $\vec{E} = \{\vec{Z}, -\vec{Z}\}$ where $\vec{E}_2$ cannot be defined. In such a case we chose $\vec{E}_2$ in an arbitrary direction orthogonal to $\vec{Z}$.

Note that aerodynamics force in the direction of $\vec{E}^1$ is the drag force, and the force in the direction of $\vec{E}^2$ and $\vec{E}^3$ is the 3D lift force. Using these bases, the aerodynamics coefficients can be written as:

$$\vec{\mathcal{C}}_F(\vec{E}) = \sum_{i=1}^{3}\sum_{l=0}^{N}\sum_{m=-l}^{l}\alpha_{lm}^i Y_{lm}(\vec{E})\vec{E}^i, \tag{7}$$

$$\vec{\mathcal{C}}_T(\vec{E}) = \sum_{i=1}^{3}\sum_{l=0}^{N}\sum_{m=-l}^{l}\beta_{lm}^i Y_{lm}(\vec{E})\vec{E}^i. \tag{8}$$

As a result, the omni-directional linear force and momentum coefficient can be expressed with $3N(N+1)/2$ discrete parameters $\mathcal{A} = \{\alpha_{lm}^i\}$, $\mathcal{B} = \{\beta_{lm}^i\}$ where $0 \le i \le 3, 0 \le l \le N, -l \le m \le l$. The choice of N is discussed in § 6.

Throughout this paper, we visualize the values of aerodynamic force coefficients and torque coefficients with arrows sampled discretely on unit spheres. Arrows of force coefficients (cyan) show $\vec{\mathcal{C}}_F$ scaled by a factor of -0.5 and arrows of torque coefficients (purple) show $\vec{\mathcal{C}}_T$ scaled by a factor of 5. The object with normalized length ($L = 1$) is rendered at the center of the sphere (orange).

4.3 Simulation

Given the aerodynamic model, we can compute the animation by integrating (1) and (2) over time. While any time-integration

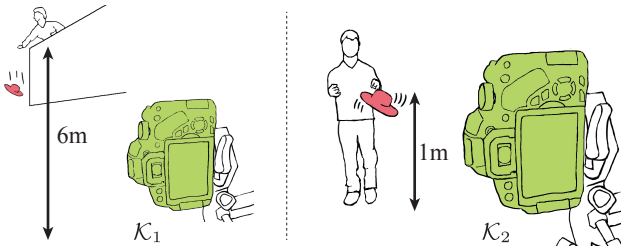


Figure 6: *Our acquisition setup. We record scenarios where either linear (left) or angular velocity (right) is dominant.*

schemes can be used to solve these equations, in this work, we use fourth-order Runge-Kutta integration (see supplemental material for more detail). We denote a state of the object at time step k as $\vec{s}_k = (\vec{y}, \mathbf{R}, \vec{v}, \vec{\Omega})^T$, and denote the time-integration function as $\vec{s}_{k+1} = f(\vec{s}_k)$.

5 Reconstruction of Aerodynamic Force

This section describes how to reconstruct the motion of a 3D rigid body from videos taken from a single viewpoint. The reconstructed motions are used to estimate the parameters for the omni-directional aerodynamic model, as discussed in the previous section.

Inputs of the motion reconstruction. All videos were captured using a Canon DSLR X7i, with an image resolution of 1280×720 at 60 frames per second (FPS). Furthermore, six to eight visual markers were placed on the object in advance, such that four of them were visible in all view directions at any time. Then we tracked the screen positions of the visible markers for each image. The user specifies the initial position and orientation of the object, as well as its mass M and inertia tensor $\mathbf{I}$.

OmniAD has two types of terms: i) velocity-dependent terms (the first terms of (5) and (6)), and ii) angular velocity-dependent terms (the second term of (6)). To estimate the parameters of both terms, two types of motion are captured: i) motions where linear velocity is dominant and ii) motions where linear velocity is small and angular velocity is dominant. For the first type of video (typically six to eight takes), the object is dropped from a height of approximately $6\mathrm{m}$, resulting in image sequences $\mathcal{K}_1$. For the second type of video (typically between four to six), the object is manually thrown in the air with a rotating motion at the height of $1\mathrm{m}$, providing the image sequences $\mathcal{K}_2$ (see Fig. 6).

Motion reconstruction. The 3D motion of the falling and rotating rigid body is determined from the reconstructed 2D marker positions in screen space. One method of reconstructing the 3D motion is to geometrically estimate the object position for the individual frames. However, because the marker positions contain small tracking errors, the estimated depth is noisy and errors are rather large. Because the object is rigid and its motion follows the physics of a rigid body, we instead use Kalman filters [Welch and Bishop 2001] to guess the 3D rigid body motion from the input marker positions. With a reasonable physics model $f(\vec{s})$ and observation function $h : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, Kalman filters are known to produce a reasonable estimation. In our case, $h(.)$ projects a 3D position to 2D, using the intrinsic camera parameters obtained with the camera calibration toolbox for Matlab [Bouguet 2000]. We used our OmniAD framework as the physics model $f(\vec{s})$. In the next paragraphs, we explain how we estimated the parameters of our model. For further implementation details about our Kalman filter, we refer the reader to the supplemental material.

Aerodynamic force reconstruction. Let us denote the reconstructed rigid body state at each frame as $\vec{s}_k^* \quad k \in \mathcal{K}_1, \mathcal{K}_2$. In the following, we use an asterisk to indicate reconstructed quantities. Given $\vec{s}_k^*$, we compute the force and torque from the example sequence using central differences applied to (1) and (2) as

$$\vec{F}_k^* = \mathbf{R}_k^{*-1} \{M(\vec{v}_{k+1}^* - \vec{v}_{k-1}^*)/(2\Delta t) - M\vec{g}\}, \quad (9)$$

$$\vec{T}_k^* = \mathbf{I}(\vec{\Omega}_{k+1}^* - \vec{\Omega}_{k-1}^*)/(2\Delta t) + [\Omega_k^*]\mathbf{I}\vec{\Omega}_k^*, \quad (10)$$

where Δt is the time interval between two frames. These forces are computed for all the frames in the videos.

Stability enforcement. From the quasi-static assumption, the aerodynamic force $\vec{F}$ causes the velocity of the object to decrease: i.e., given any configuration, the object does not obtain any kinetic energy from the surrounding air. Hence, the aerodynamic coefficient function $\mathcal{C}_F(\vec{E})$ in (5) has to satisfy $\mathcal{C}_F(\vec{E}) \cdot \vec{E} < 0$ for any direction $\vec{E}$. We also ignore the frames where $\vec{F}_k \cdot \vec{V}_k > 0$. This ensures that the acquired aerodynamic model does not produce artificial thrust forces. As discussed in § 4.1, the angular velocity damping tensor $\mathbf{D}$ has to be positive definite. We ensure this property by ignoring frames for which $\vec{T}_k \cdot \vec{\Omega}_k > 0$.

The following section describes how to obtain parameters for our omni-directional aerodynamic model, given these reconstructed forces and motions.

6 Force Space Parameter Fitting

This section describes how we identify the parameters for our proposed aerodynamic model, as described in § 4.2. The previous section explains how to reconstruct the rigid body motion and aerodynamic forces and torques from the given 2D videos. These reconstructed example data are collected, which include the object's linear velocity and angular velocity $\langle \vec{V}_k^*, \vec{\Omega}_k^* \rangle$, and the object's force and torque $(\vec{F}_k^*, \vec{T}_k^*)$. From these data, we establish the relationship between velocities and forces $\langle \vec{V}, \vec{\Omega} \rangle \rightarrow (\vec{F}, \vec{T})$, as accurately as possible. The outputs of this are the parameters used to evaluate the proposed omni-directional aerodynamic model.

Note that in our parameter-identification technique, we fit the forces instead of the trajectories of the marker positions. When identifying the parameters of elastic objects, it is common to fit displacements to determine stiffness parameters [Otaduy et al. 2012]. However, unlike elastic objects, where elastic reaction forces are determined mainly by displacements, aerodynamic forces largely depend on the object's velocity (§ 4). Moreover, even a small perturbation in the initial setting results in a significant difference in the final position. Because a rigid body inside a fluid undergoes stochastic motion, it is difficult to identify a physics model that accurately matches the acquired trajectories. The force space parameter fitting used in this paper can identify parameters when there is a strong relationship between velocity and force, even if the resulting trajectory is chaotic and too sensitive to the aerodynamic parameters.

Fitting linear force. The parameters $\mathcal{A}$ for the omni-directional aerodynamics coefficient $\vec{\mathcal{C}}_F$ are obtained from the frames capturing the object's falling sequences $\mathcal{K}_1$. For frame $k \in \mathcal{K}_1$, the aerodynamics coefficient for the direction $\vec{E}_k^*$ corresponding to the frame can be computed as $\vec{\mathcal{C}}_F^*(\vec{E}_k^*) = \vec{F}_k^*/(0.5\rho|\vec{V}_k^*|^2 L^2)$. Because the bases of the spherical harmonics are ortho-normal, we can obtain the coefficients of spherical harmonics α_{lm}^i by comput-

ing the convolution over a unit sphere S_1.

$$\alpha^i_{lm} = \int_{S_1} \left(\vec{C}^*_F(\vec{E}) \cdot \vec{E}^i\right) Y^l_m(\vec{E}) ds, \tag{11}$$

where $\vec{C}^*_F(\vec{E})$ is a coefficient for a given direction $\vec{E}$ obtained through interpolating discretely sampled coefficients at every frame $\vec{C}^*_F(\vec{E}^*_k)$. For this interpolation we use a three-nearest neighbor method where the distance measure is $|\vec{E} - \vec{E}_k|$. We compute this integration numerically over a discretized unit sphere with 32 longitudinal divisions and 16 latitudinal divisions.

Fitting damping torque. We first fit the angular damping coefficient by matching the recovered torques reconstructed from the first set of video frames, defined as $\mathcal{K}_2$, capturing the object's rotations. Note that the damping coefficient is a symmetric tensor; thus, only six independent parameters need to be computed.

$$\mathbf{D} = \underset{D_{ij}, 1\le i\le j\le 3}{arg\,min} \sum_{k\in\mathcal{K}_1} \|\vec{T}^*_k - \frac{1}{2}\rho\mathbf{D}|\vec{\Omega}^*_k|\vec{\Omega}^*_k L^5\|^2. \tag{12}$$

Fitting velocity dependent torque. After computing the damping coefficient, the parameters of the direction-dependent torque coefficient $\mathcal{B}$ are obtained. From the obtained damping coefficient $\mathbf{D}$ and (6), the amount of torque at each frame k is estimated to be $\vec{C}^*_T(\vec{E}^*_k) = (\vec{T}^*_k - 0.5\rho\mathbf{D}|\vec{\Omega}^*_k|\vec{\Omega}^*_k L^5)/(0.5\rho|\vec{V}^*_k|^2 L^3)$. Similar to the previously obtained aerodynamics force coefficient, the coefficients of SH can be computed by applying a convolution over the unit sphere.

$$\beta^i_{lm} = \int_{S_1} \left(\vec{C}^*_T(\vec{E}) \cdot \vec{E}^i\right) Y^l_m(\vec{E}) ds. \tag{13}$$

Iterative refinement. Because the aerodynamic model of the object is unknown, a Kalman filter is used to guess the best rigid body's motion to reconstruct the input trajectory. The guess made by the Kalman filter is refined iteratively, by adding the reconstructed forces and parameters of our model. First, the aerodynamic force and torque are initialized with zero (i.e., $\mathcal{A}_0 = 0, \mathcal{B}_0 = 0$), from which the first guess of model parameters is obtained $(\mathcal{A}_1, \mathcal{B}_1)$. Then, using the first guess, the Kalman filter is applied again to improve the guess of the rigid body motion to get a second guess of the aerodynamics coefficients $(\mathcal{A}_2, \mathcal{B}_2)$. In practice, we observed that the model estimation converges quickly. Fig. 7 shows how the sampled and interpolated aerodynamics force coefficient $\vec{C}^*_F(\vec{E})$ changes with respect to these iterations. Initially, the Kalman filter cannot discern the process noise from actual forces

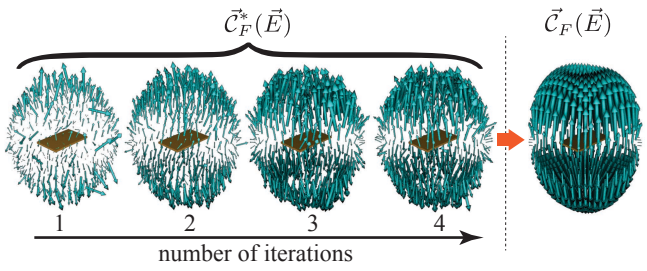


Figure 7: *Iterative estimation of aerodynamic coefficient. The acquired aerodynamics force quickly converges by iterating between Kalman filter motion estimation and parameter fitting. The spherical harmonics representation of $\vec{C}_F(\vec{E})$ further reduces noise.*

well, because it assumes no aerodynamics forces as underlying physics model. Therefore, in the first iteration, a lot of noise is present in the coefficients. However, by iterating this process, we refine the accuracy of the physics model used in the Kalman filter, and thus improve the filter's accuracy. Moreover, the SH representation further filters high-frequency noise. In all of our experiments, three iterations were sufficient.

Model resolution. We use spherical harmonics to represent the aerodynamic forces, which change significantly with respect to the airflow direction relative to the object. By changing the maximum degree N in the spherical harmonics approximation, we can adjust the level of detail in the force change with respect to the direction. Fig. 8 shows a comparison where a falling foam plate is simulated using $N = 1$, $N = 8$, and $N = 10$. The falling simulation with $N = 1$ is too smooth because it does not capture the force change with respect to direction. On the other hand, simulations with both $N = 8$ and $N = 10$ reproduce rapid velocity changes, capturing the characteristic motion of the actual sequence. In this paper we use $N = 8$, because it provides sufficient accuracy to build aerodynamically functional box kites.

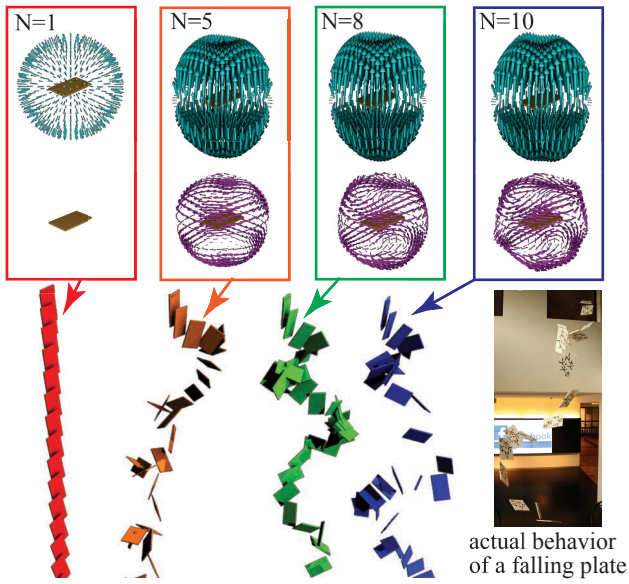


Figure 8: *Simulating a foam plate with different spherical harmonics resolutions (cyan arrows illustrate the force coefficient $\vec{C}_F$ and purple arrows the torque coefficient $\vec{C}_T$). The initial conditions are identical. Increasing the resolution gives more plausible and complex animations.*

7 Application to 3D Kite Design

Because OmniAD is based on actual measured data, it has a "true" physical meaning. Thus, it can to be used for the design and fabrication of aerodynamically functional objects. To demonstrate this, we describe how OmniAD can be used to interactively design 3D kites. A kite flies stably in air when it reaches an equilibrium of lift forces, drag forces, torque, and string tension forces. We propose an interactive kite modeling tool, called *KiteShop*, which supports the user in creating 3D kites by visualizing these quantities in real time. While the dynamics of simple plane kites have been studied before [Veen 1996; Wright 1998], our tool allows the user to design free-form original 3D box kites. Our tool also makes small on-the-fly adjustments to the string attachment position to easily achieve a state of equilibrium.

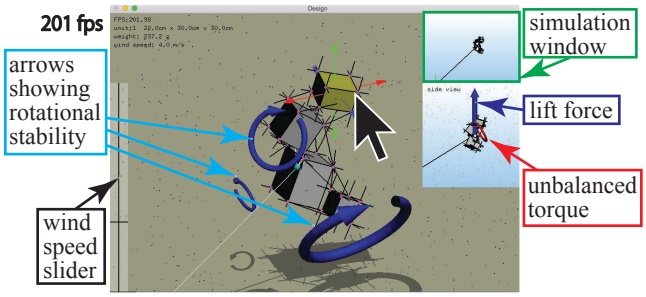


Figure 9: *Screen-shot of our kite design interface.*

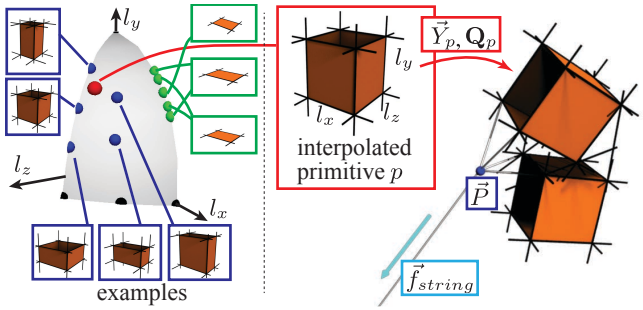


Figure 10: *Left: interpolation of example coefficients for a given box shape (red point). Normalized dimensions of box examples and plate examples are shown as blue and green points, respectively. The black points represent example shapes with no surface area, thus all coefficients are zero at these locations. Right: configuration of a kite.*

User interface. Fig. 9 shows the user interface of KiteShop. The user can assemble box primitives, which are cuboids with two open opposing faces, by rotating, translating, and scaling them in 3D. The shape of the box primitive is parameterized with its dimensions (height, depth, width) and the user can adjust these parameters interactively. Furthermore, the user can add an arbitrary 3D object, the aerodynamic properties of which are captured with OmniAD. To construct a single rigid frame structure, the user manually specifies the pairs of primitives that are connected with rods. Currently the tool does not perform structural analysis. However, if a primitive is connected to other primitives with more than three rods, it usually gives enough rigidity to the frame structure. During the user's interactive kite shape editing, the system visualizes its equilibrium and stability based on a real-time simulation (top-right window), string force and unbalanced torque (middle-right window), and stability (main window). The position to attach a string is automatically optimized during editing to achieve maximum equilibrium.

Aerodynamics parameterized primitive. We used spherical harmonic coefficients to interpolate aerodynamic properties of parameterized primitives. Fig. 10-left illustrates how a box primitive is parameterized and how we interpolate given example aerodynamic properties. The local coordinate of a box is defined such that the opening sides is on the y-axis. The longer edge of the closing face is the x-axis and the shorter edge is on the z-axis. We denote these lengths in x, y and z axis as l_x, l_y, and l_z, respectively. Because OmniAD is defined on the normalized shape $L = l_x^2 + l_y^2 + l_z^2 = 1$ and we exploit the symmetry of a box, thus a box shape can be parameterized $1/8$th of a unit hemisphere ($l_y > 0, l_x > l_z > 0$). A rectangular plate is treated as a special case of a box primitive where length along z is zero: i.e., $l_z = 0$. With this normalized dimension space, SH coefficients are blended using normalized weights, based on the inverse distance in parameter space.

Force on a kite. Fig. 10-right shows the configuration of primitives in a kite. We denote $\mathcal{P}$ as a set of primitives in the kite. Each primitive in the kite $p \in \mathcal{P}$ is rotated with $\mathbf{Q}_p$ and translated $\vec{Y}_p$ from its reference configuration. As illustrated in Fig. 3, the entire kite is then rotated against the global frame with $\mathbf{R}$. We denote $\vec{P}$ as the attachment point of the string to the kite. Force and torque at the center of gravity on a kite can be written as:

$$\vec{f}_{kite} = M\vec{g} + \vec{f}_{string} + \mathbf{R}\sum_{p\in\mathcal{P}} \mathbf{Q}_p \vec{F}_p, \tag{14}$$

$$\vec{T}_{kite} = \vec{P} \times (\mathbf{R}^T \vec{f}_{string}) + \underbrace{\sum_{p\in\mathcal{P}} \mathbf{Q}_p \vec{T}_p + \vec{Y}_p \times (\mathbf{Q}_p \vec{F}_p)}_{\vec{T}_{air}}, \tag{15}$$

where $\vec{f}_{string}$ is the tension of the string. $\vec{F}_p$ and $\vec{T}_p$ represent the aerodynamic force and torque of primitive p. Because an object in the flow predominantly affects the aerodynamic force of downstream objects [Veen 1996], the tool prohibits placing two overlapping objects in the wind direction; thus, we can ignore interference between primitives. With these forces, we can simulate kite dynamics using the equations of motion ((1) and (2)) and display them in real time.

Let a kite be in a static equilibrium, where linear velocity and angular velocity are zero ($\vec{v} = 0, \vec{\Omega} = 0$) and forces are in balance ($\vec{f}_{kite} = 0$, $\vec{T}_{kite} = 0$). The velocity of primitive p against air becomes $\vec{V} = -\mathbf{Q}_p^T \mathbf{R}^T \vec{v}_{wind}$. From (14) and linear equilibrium $\vec{f}_{kite} = 0$, we can compute the tension force $\vec{f}_{string}$, then we can compute unbalanced torque $\vec{T}_{kite}$ from (15). We visualize the direction of the tension force as the direction of string and unbalanced torque $\vec{T}_{kite}$ with a red circular arrow. For a kite to achieve static balance in the air, the tension force must pull the kite downwards $\vec{f}_{string} \cdot \vec{g} > 0$. When this condition is violated, the system visualizes the force with a red arrow.

Stability of a kite. A kite stays in the air stably when it produces restitution torque against perturbations. Such stability can be analyzed by taking the derivative of the aerodynamic torque. Specifically, a kite produces restitution torque if $(\partial \vec{T}_{kite}/\partial \vec{\Theta}) \cdot \vec{\Theta} < 0$, where $\vec{\Theta}$ is an arbitrary infinitesimal 3D vector that changes the rotation as $\mathbf{R}' = \mathbf{R}[\vec{\Theta}]$. This requires all three eigenvalues of the symmetric part of the Jacobian matrix $(\partial \vec{T}_{kite}/\partial \vec{\Theta})$ to be negative. During the user's editing, we perform an eigen decomposition of the symmetric part of the Jacobian matrix. If an eigenvalue of the matrix is positive, the kite is unstable around the axis in the direction of the corresponding eigenvector. The system visualizes the stability with three circular arrows around each eigenvector direction either in red (unstable) or in blue (stable).

Optimization of string attachment position. The torque force changes depending on the string attachment position $\vec{P}$. The system automatically optimizes this position such that the unbalanced torque $\vec{T}_{kite}$ becomes as small as possible during the user's editing. Given the torque generated from primitives $\vec{T}_{air}$ (the second term of the (15)), the string attachment position to minimize magnitude of unbalanced torque $|\vec{T}_{kite}|$ can be solved analytically from (15)

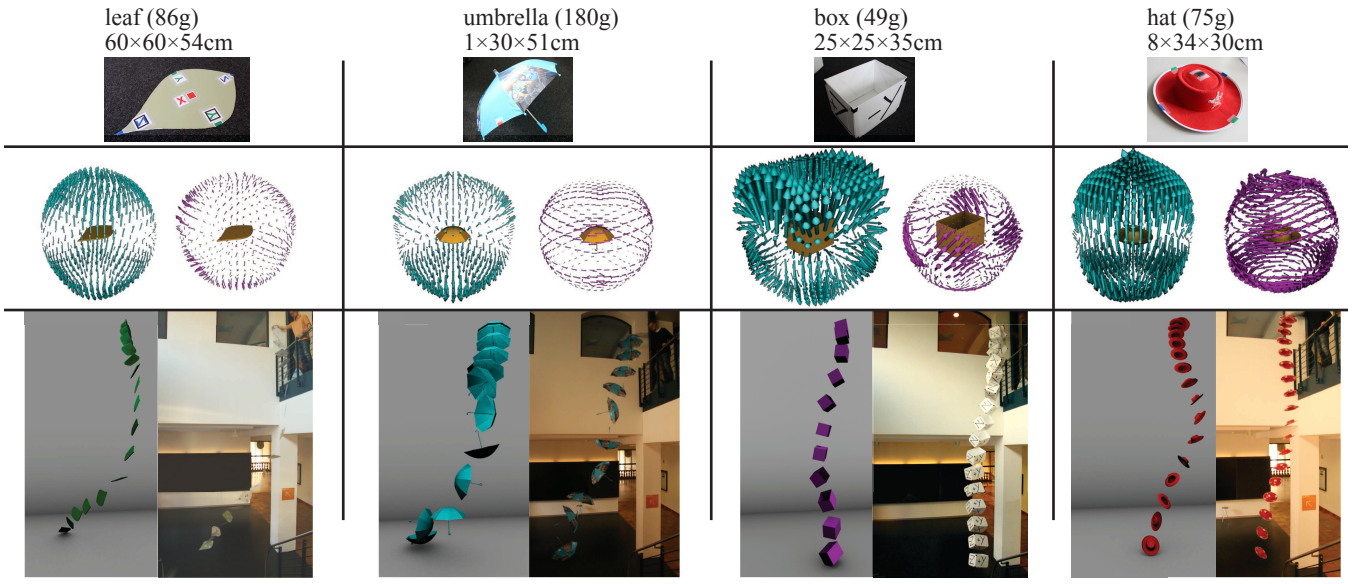


Figure 11: *Omni-directional aerodynamic coefficients of various objects. The bottom row shows side-by-side comparison of their actual and simulated falling motions.*

as a point on a line parameterized with $\gamma \in \mathbb{R}$ as:

$$\vec{P} = \vec{F}_{string} \times \vec{T}_{air} / |\vec{F}_{string}|^2 + \gamma \vec{F}_{string}. \qquad (16)$$

where, $\vec{F}_{string} = \mathbf{R}^T \vec{f}_{string}$. The choice of γ does not affect the static equilibrium, but does affect stability. Furthermore, we need to consider potential intersections of the string with primitives. Thus, we simply let the user interactively adjust γ to determine the string attachment position with the real-time stability feedback.

8 Results

We measured the aerodynamic properties for the following objects: foam plate (Fig. 2, Fig. 8 and Fig. 12), open box (Fig. 11), hat (Fig. 11), small umbrella (Fig. 11), foam heart (Fig. 1), and cardboard leaf (Fig. 11). The size of the foam plate was $30 \times 50 \times 2$ cm and had a weight of 48 g. The size of the heart shaped foam object was $5 \times 30 \times 34$ cm and had a weight of 56 g. Inertia tensors for these objects are estimated from their material densities and shapes. The dimensions of the other objects are provided in Fig. 11. It took approximately two minutes to estimate parameters after the 2D marker positions were provided. All of the acquired data is included in the supplemental material. The accompanying video shows acquired input video sequences, and generated animations of these object for various initial configurations.

As shown in Fig.11 and the video, our method is able to reproduce the characteristic behavior of the object's falling motion. For example, the foam plate glides when the velocity is low. Then it quickly gathers speed, to a point where it flips over and finally looses its momentum. The hat follows a self-induced wobbly trajectory while falling. The simulated leaves reproduce the stable spinning motions around the axis, which is predominant in the falling experiments of the cardboard leaves. The box model exhibits dynamic stability when its open side is oriented upwards. See supplemental material for comparisons between actual and simulated falling motions.

The evaluation of force and torque with our model is efficient. Thus, the computational overhead it imposes on the overall rigid body simulation pipeline can be negligible. For example, evaluating the force and torque for a single object takes less than 0.004 ms on a Macbook Pro (2.8 GHz Intel Core i7, 16 GB RAM). Our demo application simulates up to 1,000 rigid bodies at about 30 FPS including rendering time (no collision detection, or two-way coupling between objects). Our approach is thus suitable for use in real-time applications such as games or for interactive design tools.

Comparisons. Fig. 12 compares motions generated with our model against four other motions: motion in a vacuum, motion based on the Kirchhoff tensor model [Weißmann and Pinkall 2012], motion with the drag and traction model [Wejchert and Haumann 1991], and motion with simplified aerodynamics [Wu and Popović 2003]. In this experiment, we released the foam plate with no initial velocity or angular velocity. The motion with the Kirchhoff tensor shows wobbling behavior but it lacks the chaotic gliding and tumbling motion. Both the motion with the drag and traction model and simplified aerodynamics show sliding motions because resistance force is higher in the normal direction. However, these models do not reproduce the complex flipping motion because they ignore aerodynamic torque force. Our model successfully reproduces the characteristic of complex falling motions with the directional

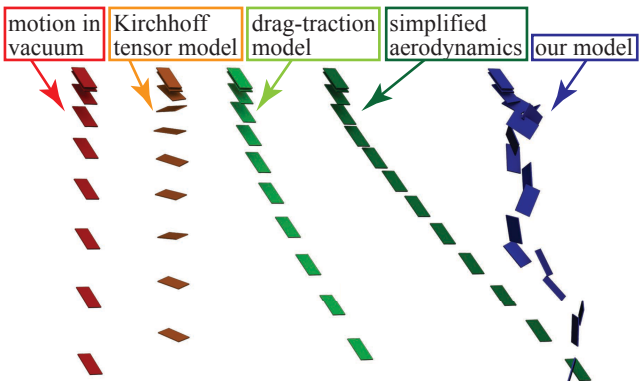


Figure 12: *Motion comparison of a falling rigid plate: Motion in a vacuum (red), motion generated using Kirchhoff tensors (orange), drag and traction model (green), simplified aerodynamics (dark green), and motion generated by our aerodynamic model (blue).*

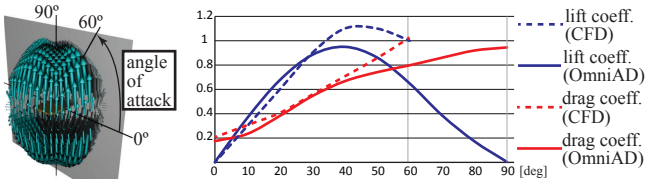


Figure 13: *Comparison of the drag and lift coefficients of the plate model against coefficients obtained from CFD simulation [Taira and Colonius 2009].*

dependent force and torque forces (see the actual falling motion in Fig. 8 and Fig. 2).

Validation. We validate obtained omni-directional aerodynamics coefficients of the plate by comparing a section of them against ground truth coefficients computed for a rectangular thin plate using high resolution CFD simulation [Taira and Colonius 2009]. Fig. 13 shows the comparison of lift and drag coefficients of the plate with the angle of attack ranging from 0 to 60 degree. Our estimation of the coefficients deviates 13% with drag coefficient and 16% with lift coefficient based on the L_1 norm. We believe this accuracy is sufficient in animation and DIY kite fabrication.

Interactive demo application. To demonstrate the effectiveness of our approach, we developed an interactive application showing falling objects (see Fig. 14). While falling motions of the many objects are simulated, the user can interactively change three parameters of the object: the fluid density (ρ), size of object (L), and the scale of mass and inertia of the object (M,I). Note that our aerodynamics model allows changing these parameters with a single set of aerodynamics coefficients.

Interactive design of 3D kites. Fig. 15 shows four 3D kites that are designed with our design tool and are also fabricated and tested in the real world. All these kites are asymmetric and dissimilar to existing traditional box kite designs. All four fabricated 3D kites fly stably in the air with a wind speed over $4\ \mathrm{m/s}$ even with fluctuations in the wind velocity, as can be seen in the accompanying video.

All of the boxes in the designed box kites have different dimensions from the originally measured box primitives, showing the effectiveness of our model and the interpolation scheme. All boxes are open in the wind direction because the aerodynamic drag force is too large when a closed surface faces the wind direction. However, the orientations of the boxes are designed carefully such that the kites achieve aerodynamic equilibrium and stability. Only with our omni-directional aerodynamic model can the user explore such boxes orientations with interactive stability and equilibrium evaluations. The omni-directional aerodynamic properties are also important in analyzing stability, because stability is determined by restoration torque when the kite's orientation is perturbed. Furthermore, based on the OmniAD model, the user can design a kite carrying an arbitrary free-form object by acquiring its aerodynamic property with a simple experiment. For example, the kite in Fig. 15-(c) carries a heart made of foam.

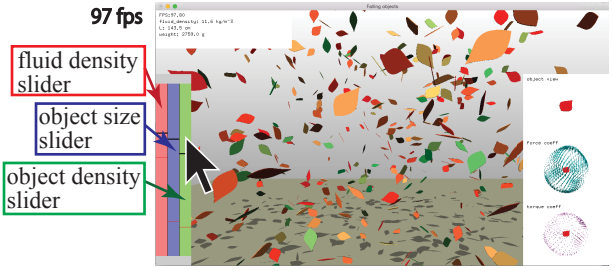


Figure 14: *Interactive parameter adjustment application. The user can adjust density of the fluid, size and weight of the falling objects.*

We constructed all kites using carbon fiber rods (diameter $2\ \mathrm{mm}$) that are connected together with 3D printed sockets. The geometry of the sockets were generated automatically using OpenSCAD such that two rods intersected with a specific angle. We put kitchen plastic wrap around the framework to cover the closed faces of each box. These materials allow very lightweight kite constructions. For example, a kite with two box primitives, which spans about $1\ \mathrm{m}$, weights about $100\ \mathrm{g}$, takes approximately $3\ \mathrm{h}$ to assemble, and costs about 50 US dollars for the construction materials.

9 Conclusion

We have presented OmniAD, an omni-directional aerodynamic model for light, rigid objects moving within air. Our model enables common users to acquire aerodynamic properties of a given object simply by recoding several videos while being dropped. OmniAD provides sufficient accuracy for creating compelling animations of lightweight objects in the air as well as for the design of real-world aerodynamic objects such as kites. Additionally, OmniAD is efficient and allows real-time simulation of hundreds of objects. Our design system *KiteShop* is based on OmniAD and provides intuitive and interactive interface for designing 3D kites. As demonstrated in the result section, we successfully estimated the omni-dimensional aerodynamic properties of a variety of objects and even fabricated several functional kites.

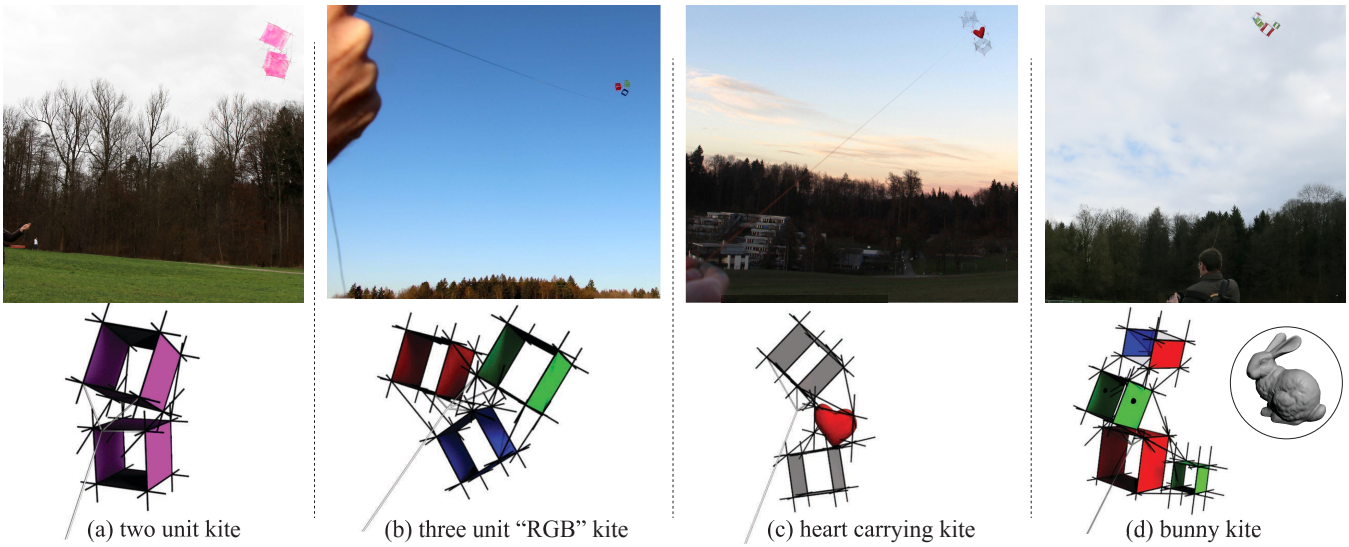
Limitations and future work. By design of our method, objects have to be lightweight, stay in the air for several seconds, and their motion trajectory needs to be affected by aerodynamics; otherwise, we cannot acquire the aerodynamic properties faithfully. Our model does not handle interference between multiple objects that are close together. For future work, we would like to investigate how interference could be handled, for example by switching aerodynamic models when an object is close to other objects. Also, we plan to investigate further use cases of our aerodynamic model, for example for interactively designing other types of functional objects such as free-form model airplanes. Finally, we think an exciting direction of future work would be to extend our framework to handle non-rigid objects such as deformable paper flying in the air, swimming fish, or flapping birds.

Acknowledgment

We thanks Haron Xie for helping us with the implementation of Kirchhoff tensor rigid-body simulation, Peter Kaufmann, Chris Wojtan, Ryan Schmidt and Tyson Brochu for valuable comments on the paper draft, Andrin Landolt and Andreas Müller for advice on experimental aerodynamics model acquisition, and Sumathi V Selvaretnam and Marcel Lancelle for their help with the kite flying experiments.

References

ABBOTT, I. H. 1959. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Dover Publications.



(a) two unit kite (b) three unit "RGB" kite (c) heart carrying kite (d) bunny kite

Figure 15: *Fabricated 3D kites*

ANDERSEN, A., PESAVENTO, U., AND WANG, Z. J. 2005. Unsteady aerodynamics of fluttering and tumbling plates. *Journal of Fluid Mechanics 541* (10), 65–90.

BÄCHER, M., WHITING, E., BICKEL, B., AND SORKINE-HORNUNG, O. 2014. Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4.

BATCHELOR, G. K. 2000. *An Introduction to Fluid Dynamics (Cambridge Mathematical Library)*. Cambridge University Press, 2.

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3.

BICKEL, B., BÄCHER, M., OTADUY, M. A., MATUSIK, W., PFISTER, H., AND GROSS, M. 2009. Capture and modeling of non-linear heterogeneous soft tissue. *ACM Trans. Graph. (Proc. SIGGRAPH) 28*, 3.

BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior.

BOUGUET, J. 2000. Matlab camera calibration toolbox.

CARLSON, M., MUCHA, P. J., AND TURK, G. 2004. Rigid fluid: Animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3, 377–384.

CEYLAN, D., LI, W., MITRA, N. J., AGRAWALA, M., AND PAULY, M. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.

CHEN, D., LEVIN, D. I. W., DIDYK, P., SITTHI-AMORN, P., AND MATUSIK, W. 2013. Spec2Fab: A reducer-tuner model for translating specifications to 3D prints. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

CHENTANEZ, N., GOKTEKIN, T. G., FELDMAN, B. E., AND O'BRIEN, J. F. 2006. Simultaneous coupling of fluids and deformable bodies. In *Proc. SCA*, 83–89.

CIGNONI, P., PIETRONI, N., MALOMO, L., AND SCOPIGNO, R. 2014. Field-aligned mesh joinery. *ACM Trans. Graph. 33*, 1.

COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

HILDEBRAND, K., BICKEL, B., AND ALEXA, M. 2012. crdbrd: Shape fabrication by sliding planar slices. *Comput. Graphics Forum (Proc. Eurographics) 31*, 2pt3, 583–592.

HULLIN, M. B., IHRKE, I., HEIDRICH, W., WEYRICH, T., DAMBERG, G., AND FUCHS, M. 2013. Computational fabrication and display of material appearance. In *Eurographics STARs*.

IHMSEN, M., ORTHMANN, J., SOLENTHALER, B., KOLB, A., AND TESCHNER, M. 2014. Sph fluids in computer graphics. In *Eurographics 2014 - State of the Art Reports*.

JU, E., WON, J., LEE, J., CHOI, B., NOH, J., AND CHOI, M. G. 2013. Data-driven control of flapping flight. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 5.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. SGP*, 156–164.

KLINGNER, B. M., FELDMAN, B. E., CHENTANEZ, N., AND O'BRIEN, J. F. 2006. Fluid animation with dynamic meshes. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3, 820–825.

LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. 2014. Build-to-Last: Strength to weight 3D printed objects. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4.

MCCRAE, J., SINGH, K., AND MITRA, N. J. 2011. Slices: A shape-proxy based on planar sections. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 30*, 6.

MIGUEL, E., BRADLEY, D., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., OTADUY, M. A., AND MARSCHNER, S. 2012. Data-driven estimation of cloth simulation models. *Comput. Graphics Forum (Proc. Eurographics) 31*, 2pt2, 519–528.

MIGUEL, E., TAMSTORF, R., BRADLEY, D., SCHVARTZMAN, S. C., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., MARSCHNER, S., AND OTADUY, M. A. 2013. Modeling and estimation of internal friction in cloth. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 32*, 6.

MORI, Y., AND IGARASHI, T. 2007. Plushie: an interactive design system for plush toys. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3.

OTADUY, M. A., BICKEL, B., BRADLEY, D., AND WANG, H. 2012. Data-driven simulation methods in computer graphics: Cloth, tissue and faces. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, 12:1–12:96.

PAI, D. K., DOEL, K. V. D., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3D objects. SIGGRAPH '01, 87–96.

PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make it stand: Balancing shapes for 3D fabrication. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. *Proc. of ACM SIGGRAPH '02 21*, 3, 517–526.

ROBINSON-MOSHER, A., SHINAR, T., GRETARSSON, J., SU, J., AND FEDKIW, R. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. (Proc. SIGGRAPH) 27*, 3.

SAUL, G., LAU, M., MITANI, J., AND IGARASHI, T. 2011. Sketchchair: An all-in-one chair design system for end users. In *Proc. TEI*, ACM, New York, NY, USA, TEI '11, 73–80.

SCHWARTZBURG, Y., AND PAULY, M. 2013. Fabrication-aware design with intersecting planar pieces. *Comput. Graphics Forum (Proc. Eurographics) 32*, 2pt3, 317–326.

SKOURAS, M., THOMASZEWSKI, B., KAUFMANN, P., GARG, A., BICKEL, B., GRINSPUN, E., AND GROSS, M. 2014. Designing inflatable structures. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4.

SONG, P., FU, C.-W., GOSWAMI, P., ZHENG, J., MITRA, N. J., AND COHEN-OR, D. 2013. Reciprocal frame structures made easy. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

STAM, J. 2009. Nucleus: Towards a unified dynamics solver for computer graphics. In *IEEE International Conference on Computer-Aided Design and Computer Graphics*, IEEE, 1–11.

STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: improving structural strength of 3D printable objects. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

TAIRA, K., AND COLONIUS, T. 2009. Three-dimensional flows around low-aspect-ratio flat-plate wings at low reynolds numbers. *Journal of Fluid Mechanics 623* (3), 187–207.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3.

UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4.

UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

UMETANI, N., KOYAMA, Y., SCHMIDT, R., AND IGARASHI, T. 2014. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4.

VEEN, H. V. 1996. *The Tao of Kiteflying: The Dynamics of Tethered Flight*, stated first printing ed. Kitelines Bookstore Llc, 3.

WANG, Z. J., BIRCH, J. M., AND DICKINSON, M. H. 2004. Unsteady forces and flows in low reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology 207*, 3, 449–460.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4.

WEISSMANN, S., AND PINKALL, U. 2012. Underwater rigid body dynamics. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

WEJCHERT, J., AND HAUMANN, D. 1991. Animation aerodynamics. *Proc. of ACM SIGGRAPH '91 25*, 4, 19–22.

WELCH, G., AND BISHOP, G. 2001. An introduction to the kalman filter. In *SIGGRAPH 2001 Cours*, 12–17.

WRIGHT, C. 1998. *Kite Flight: Theory And Practice*. Diane Pub Co, 4.

WU, J.-C., AND POPOVIĆ, Z. 2003. Realistic modeling of bird flight animations. *ACM Trans. Graph. (Proc. SIGGRAPH) 22*, 3.

XIE, H., AND MIYATA, K. 2013. Stochastic modeling of immersed rigid-body dynamics. In *SIGGRAPH Asia 2013 Technical Briefs*, 12:1–12:4.

YUAN, Z., CHEN, F., AND ZHAO, Y. 2011. Stochastic modeling of light-weight floating objects. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, 213–213.

ZHONG, H., CHEN, S., AND LEE, C. 2011. Experimental study of freely falling thin disks: Transition from planar zigzag to spiral. *Physics of Fluids 23*, 1.

ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.