ARTIFICIAL INTELLIGENCE

# Modeling, learning, perception, and control methods for deformable object manipulation



**Hang Yin*, Anastasia Varava, Danica Kragic**

**Perceiving and handling deformable objects is an integral part of everyday life for humans. Automating tasks such as food handling, garment sorting, or assistive dressing requires open problems of modeling, perceiving, planning, and control to be solved. Recent advances in data-driven approaches, together with classical control and planning, can provide viable solutions to these open challenges. In addition, with the development of better simulation environments, we can generate and study scenarios that allow for benchmarking of various approaches and gain better understanding of what theoretical developments need to be made and how practical systems can be implemented and evaluated to provide flexible, scalable, and robust solutions. To this end, we survey more than 100 relevant studies in this area and use it as the basis to discuss open problems. We adopt a learning perspective to unify the discussion over analytical and data-driven approaches, addressing how to use and integrate model priors and task data in perceiving and manipulating a variety of deformable objects.**



## INTRODUCTION

Applications of robotic manipulation of deformable objects range from manufacturing industry to medicine and service robotics (*1*–*3*). Despite its importance, the robotic manipulation of deformable objects has been historically less investigated compared to rigid object manipulation, because of the integral complexity of modeling, perception, and control related to it. Latest progress in computer graphics and machine learning has provided valuable modeling techniques and data-driven paradigms that can help to overcome some of the limitations of traditional methods in deformable object manipulation (*4*–*6*). This survey provides an insight into the basic methods for modeling of deformable objects and gives an overview of recent approaches to simulating and perceiving these. We review deformable object manipulation from a learning and control perspective and discuss open research problems in related research areas.

Existing surveys on deformable object manipulation focus on modeling and applications from industrial and domestic perspectives (*1*, *7*–*9*). The focus of these works is heavily on manipulation of objects such as ropes and clothing items. We review more recent work from a learning perspective and how it is used for perceiving and manipulating various types of deformable objects. The aim is to bridge the gap between machine learning and robotics communities, by addressing problems of model priors and the amount of data needed for training various models (see Fig. 1). Recent survey on object grasping and manipulation (*10*) identifies that physical interaction with deformable objects is one of the important open problems in the area of robotics. It further identifies that learning and data-driven approaches pave the way toward equipping robots with more advanced capabilities in terms of deformable object manipulation. Thus, apart from modeling and perception for manipulation, we also survey the work on physical modeling and simulation of deformable objects that can provide the input to learning and data-driven approaches, both in terms of perception and control. To this end, this survey presents state-of-the-art techniques facilitated by merging analytical modeling and data-driven approaches and outlines important open problems toward building robotic systems capable of flexible, scalable, and robust handling of deformable objects.

Robotics, Perception, and Learning (RPL), School of Electrical Engineering and Computer Science, Royal Institute for Technology (KTH), Stockholm, Sweden.
*Corresponding author. Email: hyin@kth.se

### Outlines

We begin with popular modeling techniques and simulators of deformable objects. On the basis of this, the survey reviews literature on perception and manipulation in the two following sections, including task solutions spanning from analytical to data-driven methods. We summarize trends on the reviewed topics and discuss open problems in the last section.

## PHYSICS-BASED MODELING OF DEFORMABLE OBJECTS

Modeling and simulating dynamics of deformable objects relies most often on geometric representations using particles or meshes (*11*, *12*). Under Newton's second law, particle or vertex motion can be described by time derivative of momentum and exerted forces $f$

$$M\ddot{x} = f \tag{1}$$

where $M$ and $x = \{x_i; i = 1,2, \ldots\}$ denote the system mass and state, respectively. The state can be represented using a finite number of particles (discrete case) or defined using a displacement function (continuous case). The motion of the entire system can be computed by integrating from an initial state $x^0$. For each particle, the evolution is defined as

$$\begin{aligned} x_i^{t+1} &= x_i^t + v_i^t \Delta t \\ v_i^{t+1} &= v_i^t + \frac{\Delta t}{m_i}(f_i^{\text{int}} + f_i^{\text{ext}}) \end{aligned} \tag{2}$$

with $\dot{x}_i = v_i$, and exerted forces explicitly divided into two parts. The component $f_i^{\text{ext}}$ sums up external contributions such as gravity and input forces, which are known for the given time step. For a constrained system, this may also include external constraint forces from, for instance, state boundary conditions. $f_i^{\text{int}} = f_i^{\text{int}}(x^t, v^t, \theta)$ captures internal constraint forces that are subject to deformation-related parameters, e.g., deformed configuration $x^t$, $v^t$, and material properties $\theta$. Modeling these internal effects has an important role

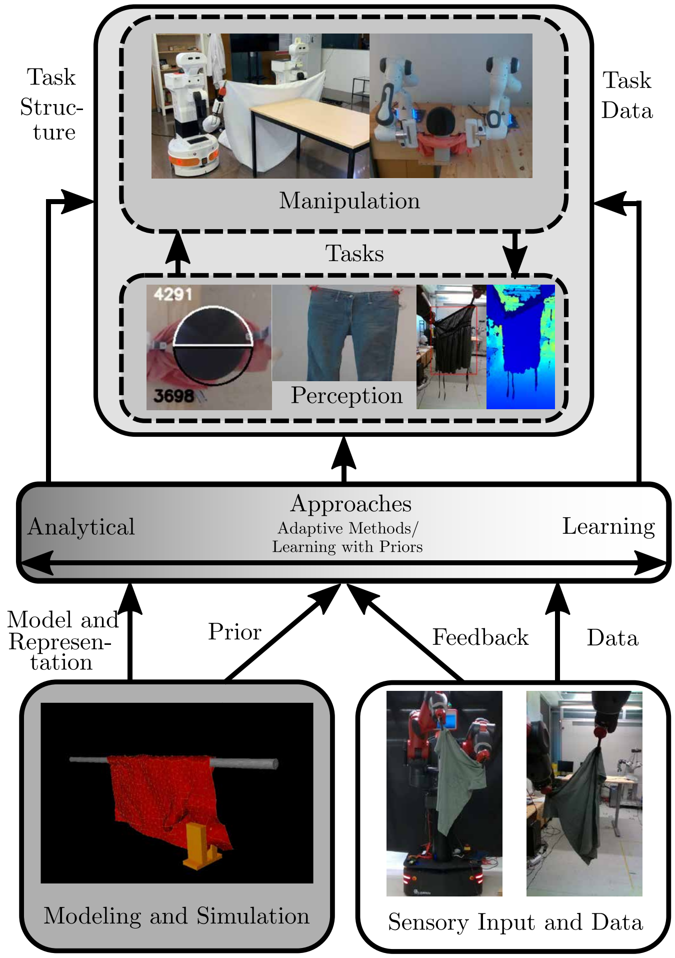


**Fig. 1. Taxonomy of deformable object manipulation.** We provide an overview of approaches ranging from analytical to learning-based methods. We address modeling and simulation of deformable objects in terms of perception, manipulation planning, and control, with the focus on learning-based methods.

in simulation of deformable objects. Once force terms are resolved, it is straightforward to perform the simulation with an explicit Euler integration, similar to Eq. 2. Other schemes such as semi-implicit integration and Runge-Kutta methods may also be used for an improved accuracy. Nevertheless, stable simulation often requires to integrate with a small time step whose size depends on the application.

A more principled scheme is implicit integration, e.g., backward Euler method. In implicit integration, the internal force term $f_i^{int} = f_i^{int}(x, v, \theta)$ depends on the future states $x^{t+1}$ and $v^{t+1}$. The integration of Eq. 2 is no longer straightforward because one needs to solve an equation to obtain $x^{t+1}$ and $v^{t+1}$. On the positive side, the simulation gains stability because the consecutive states are less deviant from dynamic constraints.

Specific examples of modeling and simulation feature different types of principles and integration schemes. These variations often trade between modeling accuracy, robustness, and computational costs. Computer graphics applications such as character or scene animation can afford an accurate offline process with a careful tuning of simulation parameters. In robotics, robustness and efficiency can be crucial for tasks involving real-time tracking, interactive

**Table 1. Overview of deformable objects modeling approaches.**

| | Advantages | Limitations | Modeling applications |
|---|---|---|---|
| MSSs | Fast | Innacurate for large deformation | Ropes (13, 14, 15) |
| | | | Fabrics (16, 17) |
| | Simple to implement | Lacking physical interpretability | Sponges (17, 18) |
| | | | Rubber spheres (18) |
| PBD | Fast and stable | Visual fidelity only | Paper (19) |
| | | | Fabrics (20, 21, 22) |
| | Supports modeling of various objects | Lacking physical interpretability | Cushion (23) |
| | | | Liquids (24) |
| Continuum mechanics (FEMs) | Can be fast (linear) | Complex and expensive to compute (nonlinear) | Rods and cables (25, 26) |
| | | | Fabrics (27, 28) |
| | Physical fidelity and interpretability | Not well integrated to rebotics simulator yet | Food (29) |
| | | | Tissues (2, 30) |

control, or learning through exploring simulated environments. Accuracy might also be paramount for safety-critical tasks such as medical applications. Bearing these in mind, the next section provides a brief review of methods that are derived from different physical models and simulators. Table 1 gives a glance of the main merits, shortcomings, and applications of existing techniques.

## Mass-spring systems

Mass-spring systems (MSSs) model deformable materials as a network. Mass is assigned to each individual vertex, and vertices are interconnected with spring edges, as illustrated in Fig. 2 (left). The internal force between a pair of vertices $i$ and $j$ can thus be determined by a spring-damping relation along the displacement direction. In implicit integration, these equality constraints are linearized around the current state so the system can be solved with, for instance, Newton-Raphson method (*11*).

MSS is simple to implement and fast to simulate. It has been applied to different types of deformable objects, for instance, by simulating a rope as a chain (*13*) or a cloth as a two-dimensional (2D) grid (*14*). The disadvantage lies in the fact that it is most suitable for simulating small deformation but not complex elastic effects with a high fidelity. In general, increasing mesh resolution cannot guarantee convergence toward the correct behavior. The network topology and spring parameters affect that as well (*11*). The spring parameters also lack a physical meaning associated to material parameters, implying a substantial tuning effort to achieve desired dynamical properties.

## Position-based dynamics

Position-based dynamics (PBD) (*11*) is a mesh-free method, modeling materials as a discrete system with particles. The simulation works as an implicit integration with internal forces derived from holonomic constraints, which might include temporary and unilateral constraints.

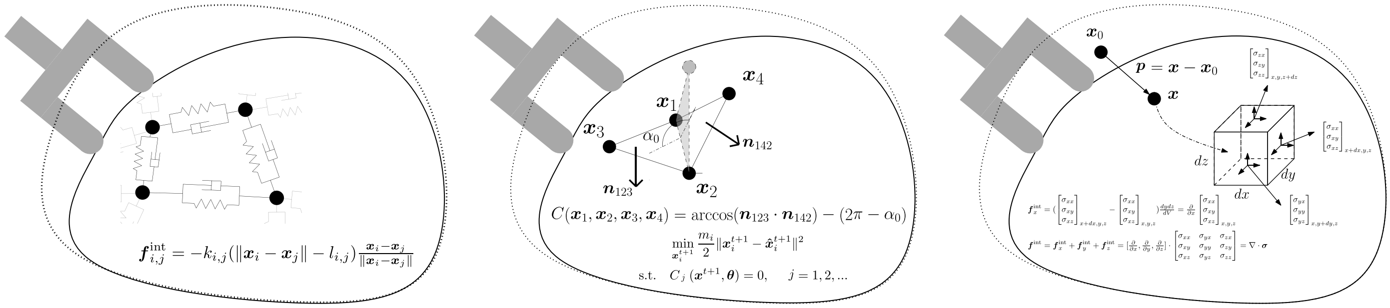


**Fig. 2. Physics-based modeling of deformable objects.** (**Left**) Discrete particles connected as a mass-spring network. (**Middle**) Discrete particles with position-based constraints: bending resistance as an example. (**Right**) Material continuum considering displacement and internal stress of infinitesimal elements as tensor fields.

$$\begin{aligned} x_i^{t+1} &= x_i^t + \Delta t v_i^{t+1} \\ v_i^{t+1} &= v_i^t + \frac{\Delta t}{m_i}\left(f_i^{\text{ext}} - \sum_j k_j \nabla_{x_i^{t+1}} C_j(x_1^{t+1}, x_2^{t+1}, \ldots, x_n^{t+1})\right) \end{aligned} \tag{3}$$

where $k_j$ denotes the stiffness of the $j$th constraint $C_j(x_1, x_2, \ldots, x_n) = 0$, $j = 1,2, \ldots$. Unlike MSSs, PBD directly updates $x_i^{t+1}$ and then computes $v_i^{t+1}$ with $v_i^{t+1} = (x_i^{t+1} - x_i^t)/\Delta t$. As pointed out in (*12*), the future state $x_i^{t+1}$ is the solution of an optimization problem whose constrains are given with Eq. 3 with $k_j \to \infty$. PBD approximately solves the optimization by iteratively projecting candidate position $\hat{x}_i^{t+1}$ to the constraint manifolds, e.g., with a Gauss-Seidel method. The elastic behavior hence depends on constraint parameters $\theta$ and how strictly these constraints are enforced by $k_j$ and projection iterations. Figure 2 (middle) depicts the general principle with an example of bending constraint. Various constraints have also been proposed to model rich effects including volume preservation, frictional contact, or strain under a continuum formulation (*12*).

PBD features fast and fully controlled simulation with improved stability. The method allows to conveniently incorporate various constraints and to drive the system by prescribing boundary conditions. It also demonstrates a great versatility to model plastic deformation, fluids, and rigid-body dynamics (*15*). In robotics, this can be used to simulate manipulators, deformable objects, and their interaction in a unified model. Methods based on spring forces, on the other hand, tend to face difficulties in handling a highly stiff system like articulated rigid links.

The main limitation is that PBD does not accurately simulate force effects, and the system stiffness depends on the size of integration time step. A resolution using constraint compliance has been proposed in a more recent extension (*16*). Nevertheless, PBD generally relies on impulse updates and provides visually plausible (but not always physically realistic) simulations. Moreover, it is not always easy to find a meaningful physical interpretation of some of the parameters. Whereas particles can be coupled by constraints in an intuitive manner, drawing a connection from simulation to physical parameters, such as material modulus, is less obvious. This implies more tuning efforts to realize expected effects. Also, parameter identification based on this type of model can only provide a qualitative material characterization (*17*), similarly to MSSs.

## Continuum mechanics

Continuum mechanics offers a more physically accurate description of modeling material deformation in a continuous domain. A state $x_t$ is defined as a displacement function over material coordinates. To characterize material deformation, the considered displacement is often relative to an initial configuration $x_0$ called rest shape. The gradient of this displacement field reflects distortion of original element geometries as is shown in Fig. 2 (right). As a result, the conservation of momentum (Eq. 1) holds for each domain element

$$\rho\ddot{x} = f^{\text{int}} + f^{\text{ext}} = \nabla \cdot \sigma + f^{\text{ext}} \tag{4}$$

where $\rho$ indicates density and force terms are now body forces applied at a unit volume. $\sigma$ denotes a stress tensor that is a symmetrical matrix. Taking a dot product with the gradient operator, the stress term captures internal elastic effects as is shown in Fig. 2 (right) for a short derivation.

A constitutive model defines the relation between stress and a measurement of material deformation. Hookean materials assume a linear relation

$$\sigma = E\epsilon \tag{5}$$

namely, material linearity. E is a coefficient matrix, e.g., $6 \times 6$ in the 3D case. For isotropic materials, it is fully determined by two scalars: (i) Young's modulus that describes material stiffness and (ii) Poisson's ratio, a ratio of lateral and longitudinal strain. $\epsilon$ denotes a strain tensor and also a symmetrical matrix and depends on the deformation gradient of displacement field

$$\epsilon = \nabla p + (\nabla p)^T + (\nabla p)^T \nabla p \tag{6}$$

with $p = x - x_0$, called Green-Lagrange strain. As an approximation of it, Cauchy strain neglects the last quadratic term to obtain a linear dependency on the displacement, assuming a geometry linearity. Hookean material assumption is often limited to modeling small deformation around the rest shape. More advanced constitutive models can design an energy term that penalizes the amount of deformation, deriving nonlinear hyperelastic relations such as Neo-Hookean model (*18*).

Simulating such a system boils down to solving the partial differential Eq. 4. The original form can only be effectively solved for simple and small-scale problems, e.g., 1D string. For general cases, finite element method (FEM) is a standard computational method for solving such continuum mechanics problems. The central idea is discretizing the material domain as a mesh consisting of a finite number of geometry elements, e.g., tetrahedra. The global consistency is enforced by setting constraints on the element boundaries and approximating the original physical fields as a linear combination of basis functions. Within each element, a weak form of the governing equation can be converted to an ordinary differential equation that is easier to solve with numerical integration. Similar to MSSs, implicit integration requires to solve the internal force term with a linearization at each time step. This can be rather complicated for FEM with a general constitutive model. Linear FEM, assuming both material and geometry linearity, allows for an offline precomputation that leads to efficient simulation. The Cauchy linear strain is, however, a poor approximation to the rotational component of displacement field. Corotational linear FEM mitigates this by extracting the rigid rotation matrix and using it in a stress-strain relation with the rotational component compensated.

Modeling deformable objects as continuum materials yields more accurate simulation and hence suits applications with precision as top priority. Model parameters are also founded with a clear physical interpretation. Computing frameworks such as FEM are broadly used and are versatile to simulate various deformation effects. The computation and implementation are more involved in comparison to particle-based methods. Linear FEM is often used in robotics when the accuracy under a large deformation is not concerned. Nonlinear computing models, in general, find it difficult to achieve a real-time performance, except when using an explicit integration scheme (*2*) or a dedicated implementation with hardware acceleration (*19*).

### Physics-based simulators

Several simulators have been developed on the basis of the above theoretical basis and provide development environments in graphics, computer vision, robotics, and control communities. We briefly mention most widely used ones focusing on the modeling techniques and use in robotics.

Simulation Open Framework Architecture (SOFA) is an open source framework, developed for tissue modeling and surgical interaction (*20*). It adopts a modular design to host customized solvers for different mechanical objects, constraints, and collision geometries. SOFA provides built-in mass-spring models. It also includes a variety of FEM models, such as linear, corotational linear, and Neo-Hookean models, for the precision in medical applications. The framework has been used in motion planning (*21*), tracking deformable objects (*37*, *38*), and vision-based tip force estimation (*24*, *25*).

PhysX is an open-sourced product of NVIDIA used in game engines. It features cloth modeling based on PBD (*26*), allowing the simulation of rigid and cloth-like items without the need of bridging different simulators as in (*27*). With an extension on frictional contact handling, PhysX has been used for various tasks in human dressing, such as haptics and force prediction (*28*, *29*). It is also used as a forward dynamics model for planning dressing motion (*3*).

MuJoCo is a proprietary engine that has been used for modeling rigid-linked characters (*30*) or as the environment backend of reinforcement learning (RL) benchmarks (*31*). MuJoCo features a convex soft contact model and hence allows for optimization methods with a higher convergence rate (*32*). It takes a constraint-based perspective to solve the interaction forces. Thus, it shares some similarities to PBD, and it may be used to model deformable objects. Because this simulator originally targeted modeling of articulated robots, it provides useful interfaces for expediting robot modeling and learning algorithms. Its deformable dynamics modeling has been applied in learning a cloth folding primitive (*33*) and rope manipulation planning (*6*).

Bullet is an open source library for collision detection and multibody simulation (*34*). It builds upon PBD to simulate deformable bodies and their interaction with other world entities. In the recent development, additional models such as corotational and Neo-Hookean materials are also featured. Bullet also provides convenient interfaces for robot modeling and machine learning (*34*). Existing research uses the simulator for deformable dynamic tasks, including object tracking (*35*, *36*), sim-to-real transfer (*37*), learning manipulation (*38*), and assistive dressing in simulation (*39*).

## PERCEPTION

Control and manipulation of deformable objects depend on rich and robust perception using visual, force, tactile, and range sensing. As for rigid objects, deformable object perception tasks range from state estimation to segmentation, tracking, recognition, and classification. In case of rigid objects, state is commonly represented with a set of features/templates or 6D pose/velocity if the model of the object is available. However, representing the state of a deformable object is one of the open challenges, and the solution is usually dependent on the application. Examples include various types of mesh and template models that quantitatively or qualitatively describe the state and may also include parameters such as elasticity and plasticity. Thus, defining representations that generalize over object types, materials, and applications remains an important open problem.

In terms of sensor type, a single camera may be used for item recognition, image and scene segmentation (*40*, *41*), and visual servoing (*42*). Stereo and depth cameras are used for extracting exact 3D geometry and texture, as well as tracking of deformation (*17*, *23*, *35*, *43*, *44*). Force and tactile sensors are central for estimating material parameters through physical interaction (*45*–*48*).

Table 2 overviews recent works on deformable object perception, and Fig. 3 overviews perception tasks covered in this section. As illustrated in Table 3, we start by defining the problem of state estimation and how it is tracked. Then, we discuss challenges in estimating material properties and lastly summarize the work on detection, segmentation, and classification of deformable objects, given that these tasks are closer to the actual applications considered.

### Representation and state estimation

State estimation of a deformable object $x$ can be seen as an optimization problem based on observations $o$ and object representation $\mathbf{M}(\cdot)$

$$\begin{aligned} x^* &= \arg\min_x \left\| o - \mathbf{M}(x) \right\| \\ x &\in \text{ObjectStates} \end{aligned} \tag{7}$$

Figure 4 shows different types of representations, ranging from state templates to detailed mesh models commonly used in simulation. Below, we discuss approaches and challenges related to these.

#### *Template-based approaches*

A state template is used often to denote an image, a set of images, or some 3D representation generated by a simulator or a mesh model that is used to represent an object in a particular state (see Fig. 4,

**Table 2. Summary of main literature on sensing and perception.**

| | Sensors | | | Model or simulation data | | | Main features | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Vision 2D** | **3D** | **Force** | **Geom/MSSs** | **PBD** | **Continuum** | **Geom** | **Texture** | **Temporal** |
| **State estimation** | | | | | | | | | |
| Cusumano-Towner *et al.* (*50*) | | ✓ | | ✓ | | | ✓ | | ✓ |
| Miller *et al.* (*40*) | ✓ | | | ✓ | | | ✓ | | |
| Elbrechter *et al.* (*36*) | ✓ | | | | ✓ | | | ✓ | |
| Schulman *et al.* (*35*) | | ✓ | | ✓ | | | ✓ | | |
| Li *et al.* (*49*) | | ✓ | | | ✓ | | ✓ | | |
| Zollhöfer *et al.* (*19*) | | ✓ | | ✓ | | ✓ | ✓ | | ✓ |
| Newcombe *et al.* (*55*) | | ✓ | | ✓ | | | ✓ | | ✓ |
| Doumanoglou *et al.* (*41*) | ✓ | | | ✓ | | | ✓ | | |
| Leizea *et al.* (*2*) | | ✓ | | | | ✓ | ✓ | | |
| Petit *et al.* (*23*) | | ✓ | | ✓ | | ✓ | ✓ | | |
| Tang and Tomizuka (*56*) | | ✓ | | ✓ | | | ✓ | | ✓ |
| **Parameter identification** | | | | | | | | | |
| Frank *et al.* (*46*) | | ✓ | ✓ | | | ✓ | ✓ | | |
| Boonvisut *et al.* (*18*) | | ✓ | ✓ | | | ✓ | | ✓ | |
| Caldwell *et al.* (*60*) | | | | | | ✓ | | | |
| Bodenhagen *et al.* (*59*) | | ✓ | | | | ✓ | ✓ | | |
| Wang *et al.* (*61*) | | ✓ | | | | ✓ | ✓ | | |
| Giiler *et al.* (*17*) | ✓ | | | | ✓ | | | | ✓ |
| Langsfeld *et al.* (*58*) | | | ✓ | | | ✓ | | | |
| **Recognition and detection** | | | | | | | | | |
| Maitin-Shepard *et al.* (*74*) | | ✓ | | | | | ✓ | | |
| Ramisa *et al.* (*43*) | | ✓ | | | | | ✓ | ✓ | |
| Lui and Saxena (*64*) | ✓ | | | ✓ | | | ✓ | | |
| Twardon and Ritter (*76*) | | ✓ | | | | | ✓ | | |
| Doumanoglou *et al.* (*41*) | | ✓ | | | | | ✓ | | ✓ |
| Kampouris *et al.* (*72*) | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| Yuan *et al.* (*73*) | ✓ | | ✓ | | | | | | |
| Koganti *et al.* (*69*) | | ✓ | | | | | ✓ | | ✓ |
| Erickson *et al.* (*29*) | | | ✓ | | ✓ | | | | ✓ |
| Sun *et al.* (*44*) | | ✓ | | | | | ✓ | ✓ | |
| Hu *et al.* (*66*) | ✓ | | | | | | | | |
| De Gregorio *et al.* (*67*) | ✓ | | | | | | ✓ | | |
| Seita *et al.* (*75*) | | ✓ | | | | | | | |
| Grannen *et al.* (*65*) | ✓ | | | ✓ | | | ✓ | ✓ | |

left). State templates and various types of reduced-state representations are commonly used in applications where a qualitative state representation is sufficient. For example, (*14*) uses state templates based on mass-spring model for clothes handling, and (*49*) matches Kinect data to a database of simulated templates from Maya simulator. The state estimation process relies on vertex distances (*14*) or a calibrated Hamming distance (*49*) for a volumetric representation. Work in (*50*) proposes a reduced-state representation from the garment contour (see Fig. 4, middle left) used for folding (*51*). State representation of elastic rods is studied in (*52*) and used for planning deformation actions (*53*). The configurations of the elastic rod are treated as solutions of a geometric optimal control problem.

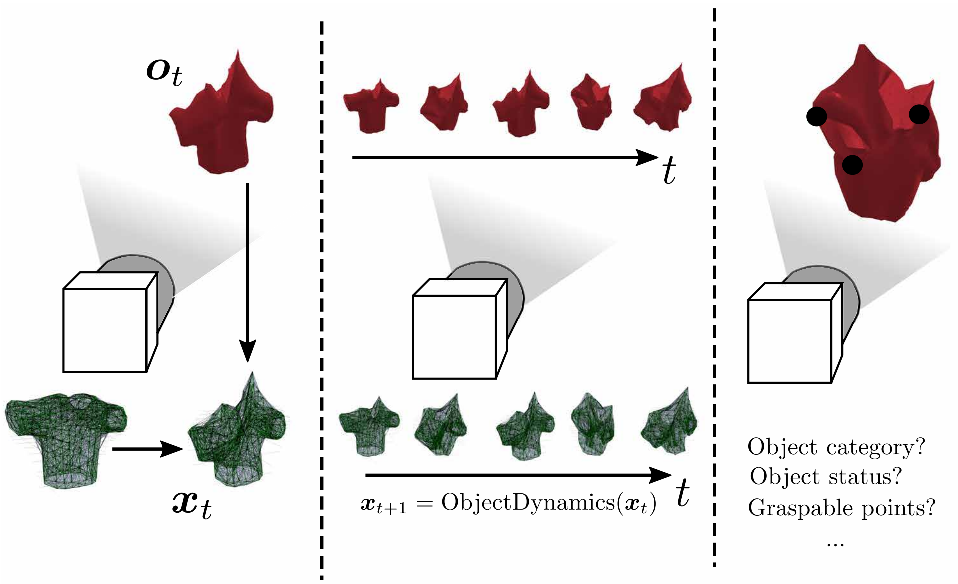


**Fig. 3. Deformable object perception tasks.** From left to right, existing research often relies on a decreased assumption of knowing the full object model. (**Left**) Estimating and tracking object state, e.g., vertex position. (**Middle**) Estimating object dynamics from sensory data. (**Right**) Recognizing object attributes and feature points.

***Geometric models***

Whereas template-based models can be seen as a qualitative representation, geometric approaches are used to encode the connectivity and spatial transformation between model vertices. For example, (*40*) relies on an approximate polygonal representation of cloth configuration for folding applications. A model-based optimization approach is used for visually inferring the class and pose of a spread-out or folded clothing article from a single image. More advanced representations add an extra energy term to Eq. 7, which can be used together with a canonical template for registering deformable surface (*54*) or facial expressions via a multistage coarse-to-fine mesh approach (*19*). If a canonical template is not available, it can be iteratively updated (*55*).

***Physics-based models***

Physics-based simulators can enable state estimation and tracking by providing simulated deformation and motion models of deformable objects. (*36*) tracks paper folding using Bullet, tackling the tracking of nonelastic bending of paper sheets during manipulation. Bullet engine is also used in (*35*) to track a range of deformable objects, including a rope, cloth, and sponges. The deformation dynamics of objects are modeled as MSSs. (*56*) uses coherent point drift for point registration. This work also proposed a method that preserves local and global structures of the tracked object, demonstrating more robust tracking of ropes and T-shirts under occlusions. Corotational linear FEM is used in (*22*) to track pizza dough, at the rate of 35 Hz. (*2*) addresses medical applications and uses a nonlinear FEM to simulate sponge and organs, achieving high-accuracy and real-time performance. Several works address also multiobject tracking in various interaction scenarios (*23*, *57*).

## Parameter identification

Material properties of deformable objects, such as Young's modulus and Poisson's ratio, are commonly estimated through an interactive process where a robot actively exerts forces against the object (*46*, *58*). The estimation process may be formulated as

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \sum_t \lVert o_t - \mathbf{M}_t(\hat{x}_t) \rVert \\ \hat{x}_{t+1} &= \mathrm{ObjectDynamics}(\hat{x}_t, f_t, \theta) \end{aligned} \tag{8}$$

where $\hat{x}$ is the predicted state that depends on object parameters such as material properties $\theta$ and applied forces $f$. Solving this optimization problem requires a backward simulation of the dynamics model. Given the complexity of deformable object dynamics, analytical gradients are rarely derived, and most works often rely on finite-difference approximations (*24*, *46*) or evolution strategies (*59*). These are computationally expensive and commonly performed offline.

In terms of modeling simple string-like objects, one commonly relies on representing elastically connected links (*60*) or beam equations (*59*). FEM-based physics is frequently used in parameter identification for more complex 3D objects. An early work in (*46*) uses linear FEM to identify Young's modulus and Poisson's ratio of household objects. Force data are collected by poking objects, and object meshes are obtained by fusing multiview point cloud data from a time-of-fly camera. In the manipulation domain, (*18*) addresses estimation of tissue parameters and forces at the tip of a robotic gripper by simulating a Neo-Hookean material model. In more recent works, the parameters of a linear FEM model are estimated to predict forces from visual input (*24*) or estimate spatially varying stiffness (*58*). In the graphics domain, (*61*) puts focus on heterogeneous elastic parameters with no assumption of a uniform Young's modulus.

PBD have also been exploited in identifying material properties. (*17*) identifies position constraint parameters by poking the different areas of a foam and using an optical flow algorithm to track the deformation. Shape-matching constraints are used as a geometry-based measurement of deformation in (*62*), where the parameter that controls volume preservation is estimated and compared to the Poisson's ratio identified from a FEM model.

## Application-specific representations

A large portion of the work on deformable objects in robotics is application specific. Thus, object representation and state estimation are commonly defined with a specific application in mind. Sorting of garment items may rely on a classification approach, whereas garment folding may need a representation that defines good grasping points on the item. Data-driven methods are frequently used, on the basis of visual or multimodal sensing, and are tied to the type of the embodiment used to execute the task. In folding, approaches commonly use some canonical representation and, on the basis of an offline trained model, generate a set of actions that result in some desired folded state (*63*). (*64*) addresses rope untangling and models ropes as a chain, learning a function to score the validity of various rope configurations. More recent work (*65*) uses the algorithm to label key point prediction to untangle ropes

**Table 3. Overview of deformable objects perception approaches.**

| | Targets | Challenges | Strategies in the literature |
|---|---|---|---|
| Model-based | Estimate and track state | High-dimensional state, partial view of objects and complex dynamics | Limiting estimation to predefined states (templates) (*14*, *130*) |
| | | | Accounting for geometry relations of object vertices (*40*, *55*) |
| | | | Filtering with physical simulator prediction (*24*, *35*, *56*) |
| | Identify material parameters | Accurate state registration, complex dynamics, and expensive gradient evaluation | Using FEM simulation and finite difference (*18*, *61*) |
| | | | Gradients from automatic differentiation (*93*) |
| Data-driven | Detecting object category or parts | Highly task-dependent, vast state space, and nontrivial correlations | Labeling and learning from large dataset (*131*) |
| | | | Merging application-specific features (*43*, *44*) |
| | | | Using multiple modalities (*72*, *73*) |

with tight knots. In medical applications, examples include thread tracking for suturing using a spline representation, retrieved from images with convolution neural networks (CNNs) (*66*). Industrial applications address problems such as insertion of electric wires into switchgears (*67*). In dressing scenarios, topological coordinates are used to describe the limb and sleeve relation, e.g., the extent of an arm being covered by the sleeve (*68*). The topological coordinates are retrieved from landmark positions, which can be inferred from depth images of garments (*69*).

Problems such as garment-type classification are commonly defined as a multiclass classification problem. (*41*) relies on large real datasets of images of real garments for both categorization and detection of grasping points. The method uses a robot to regrasp hanging garments, generating several views of the same garment to facilitate the classification problem. Recent work analyzes depth images by building a bottom-up feature hierarchy from curvatures and wrinkles to local patches. Such representations are effective for garment categorization and grasp synthesis (*44*). Tactile information is also used for classification of objects of different elasticity (*48*), characterization of food items (*70*), and the detection of failure events in dressing (*71*).

In (*72*), multiple sensor modalities and learning models are combined for identifying garment attributes, such as garment category, texture pattern, and fabric type. (*73*) associates visual and tactile modalities using GelSight, an optical-based tactile sensor. CNNs are used to encode the tactile information and the visual appearance of fabric items. Given the association, one can infer touch patterns from vision and vice versa. Learning-based approaches are also used for inferring interaction force in a dressing scenario (*29*). Using synthetic data, the work trains recurrent neural networks to predict forces applied to a mannequin's arm, given velocity and wrench measured at the grasping point. The prediction is used in model-predictive control for minimizing anticipated force exertion in dressing tasks.

Grasp synthesis for deformable objects requires detection of suitable grasping points. Early works on folding detects corners for grasping (*74*). Local geometry and appearance features are combined to determine grasping patches (*43*). A recent work (*75*) demonstrates bed making by integrating deep transfer learning and grasping. Tasks beyond grasping and folding include hanging of garments, dressing people, filling grocery bags, etc. Detecting relevant and functional parts, e.g., sleeves or collars for clothing items and bag handles, is necessary to represent such tasks. Typical approaches include skeletons or graph-based methods to locate the opening of a cap or garment (*76*).

In summary, the work on perceiving and representing deformable objects in robotics is rather broad, and there are no general solutions that apply to the various types of objects a robot may need to interact with. The reported works are often application oriented, and there are still many open challenges in terms of developing fundamental methods for representation and estimation. The most promising research directions point toward addressing multimodal perception, integration of analytical and data-driven methods, and development and use of simulators for data generation, model evaluation, and benchmarking.

## MANIPULATION

In terms of deformable object manipulation and Eq. 2, the goal is to find the actuation force $f_{\text{ext}}$ or motion at actuating points given some task objective. Similarly to perception, the reported works are rather broad and often application oriented. To this end, we first review works that address specifically planning for and control of deformable objects, discussing approaches that either rely on physics to define the relation between the state and action or use some specific heuristics. We then continue with data-driven approaches that learn the desired manipulation from examples, using both real and simulated data. We overview the taxonomy of this section in Table 4 and summarize representative literature in Table 5.

### Task and motion planning

Let us consider a task involving multistep decision-making, e.g., folding a clothing item. Typically, the problem can be formulated as an optimization problem

$$\begin{aligned} \overset{*}{x}_{0:T}, \overset{*}{u}_{0:T-1} &= \underset{x_{0:T}, u_{0:T-1}}{\arg\min}\, \mathcal{J}_{u_{0:T-1}}(x_{0:T}) \\ x_{t+1} &= \text{SystemDynamics}(x_t, u_t) \end{aligned} \tag{9}$$

with $\mathcal{J}(\cdot)$ indicating the total cost incurred by executing the planned trajectory $x_{0:T}, u_{0:T-1}$ over the time horizon $T$. We slightly abuse notation by referring $x$ to the entire system state, which may contain

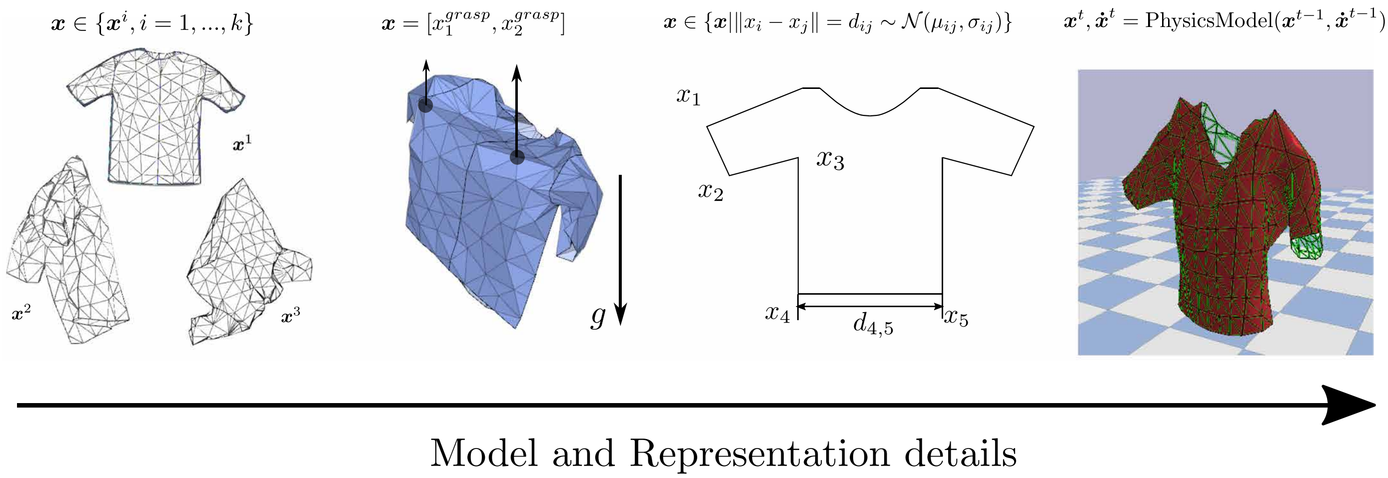


**Fig. 4. Models and representations for tracking object state.** From left to right with increased details: (**Left**) Enumeration of state templates. (**Middle**) Left: Reduced representation, e.g., grasping points on objects under gravity. Right: Geometry model, constraining vertices with geometry or energy functions. (**Right**) Physics-based models.

**Table 4. Overview of deformable objects manipulation approaches.**

| | Targets | Approaches | Challenges | Strategies in the literature |
|---|---|---|---|---|
| Model-based | Multip-step actions, e.g., pick and place | Searching action or state trajectories | Complex dynamical models, efficient exploration of high-dimensional states | Using simulators (*3*, *21*, *77*) |
| | | | | Task representation (*40*, *53*) |
| | | | | Learning dynamical models and representations (*5*, *6*, *89*, *90*) |
| | Reactive actions, e.g., shape control | Visual servoing and inverse dynamics control | Less dependent on models but only locally valid | Informative design of controllers (*95*, *103*) |
| | | | | Adaptive estimation from online data (*104*, *105*) |
| Data-driven | Complex skills, e.g., dressing | Imitation learning and RL | Expensive to acquire and label large dataset from real robot interactions | Efficient representations (*68*, *119*) |
| | | | | Task demonstrations (*116*) |
| | | | | Simulation data (*33*, *37*) |

the configuration of deformable objects, robot manipulators, and scene parameters. The system dynamics play the central role in reasoning about the effects of sequential control actions $u_{0:T-1}$ when searching for an optimal plan. In relation to this, a nominal model is commonly used to describe the state transition at both physical and task levels. Most works on manipulation focus on the motion of robots and objects, relying on a continuous representation and deformable object physics described in the "Modeling" section. The nominal model can also express high-level task dynamics, such as a sequence of grasping, moving, and releasing steps to complete folding. In that case, the transition can be rule based, relying on discrete state and action variables. The formulation of Eq. 9 is applicable to both rigid and deformable object manipulation. Consequently, algorithms commonly used for rigid body manipulation, such as trajectory optimization and sampling-based motion planning, are also used for deformable objects.

### *Shooting in the action space*

An approach to solving Eq. 9 is to apply some example action sequence and then update it according to the evaluated costs of predicted trajectories (see Fig. 5, left). This constitutes an inverse simulation problem analogous to parameter identification in Eq. 8. In general, the dimensionality of the action space is notably smaller than that of the object state, and searching in the action space can be more efficient. As per nominal models, physics-based simulations are frequently used. High-fidelity models require tuning simulation parameters or considering dynamic effects such as friction (*77*, *78*). Shooting methods on these models are mostly offline because accurately simulating the effects is expensive and closed-form gradients are often cumbersome to derive.

Works that focus on planning point-to-point movement actions to bring objects to a desired state include flipping a pizza (*79*), folding a cloth (*77*), or manipulating elastic rings (*21*). Various deformable dynamics models are explored, such as MSSs (*79*), the Maya cloth simulator (*77*), and hyperelastic FEM in SOFA (*21*). Most of these works validate the plan in simulation with some exceptions (*77*).

Grasping and in-hand manipulation of deformable objects are addressed by planning squeezing motions for lifting household objects such as vegetables and sponge-like objects (*45*, *78*). These works assume force-closure grasps and contact models between the fingertips and objects whose deformation is characterized by a linear FEM model (*78*) or nonlinear MSSs (*45*). Several computer

**Table 5. Summary of main literature on manipulation planning, control, and learning.**

| | Dimension | | | Sensors | | Manipulators | | Model/SimWorld | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1D** | **2D** | **3D** | **Vision** | **Force** | **Single** | **Multiple** | **Heuristic** | **MSSs** | **PBD** | **Continuum** |
| **Planning** | | | | | | | | | | | |
| Ramirez-Alpizar *et al.* (*79*) | | ✓ | | | | ✓ | | | ✓ | | |
| Miller *et al.* (*40*) | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| Bretl and McCarthy (*53*) | ✓ | | | | | | ✓ | | | | ✓ |
| Li *et al.* (*77*) | | ✓ | | | | ✓ | | | | ✓ | |
| Lin *et al.* (*78*) | | | ✓ | | | | ✓ | | | | ✓ |
| Satici *et al.* (*132*) | | ✓ | | | | ✓ | | ✓ | | | |
| Shah and Shah (*83*) | ✓ | | | | | | ✓ | | | | ✓ |
| Bai *et al.* (*27*) | ✓ | ✓ | | | | | ✓ | | | | ✓ |
| Zaidi *et al.* (*45*) | | | ✓ | | ✓ | | ✓ | | ✓ | | |
| Kapusta *et al.* (*3*) | | ✓ | | ✓ | ✓ | ✓ | | | | ✓ | |
| Wang *et al.* (*91*) | ✓ | | | ✓ | | ✓ | | | | | |
| McConachie *et al.* (*84*) | ✓ | ✓ | | | | | ✓ | | | ✓ | |
| Yan *et al.* (*89*) | ✓ | | ✓ | | | | ✓ | | | | |
| Lippi *et al.* (*90*) | | ✓ | | ✓ | | ✓ | | | | | |
| **Control** | | | | | | | | | | | |
| Das and Sarkar (*96*) | | ✓ | | | | ✓ | | ✓ | ✓ | | |
| Berenson (*95*) | ✓ | ✓ | | | | | ✓ | ✓ | ✓ | | |
| Kruse *et al.* (*102*) | | ✓ | | ✓ | ✓ | | ✓ | ✓ | | | |
| Navarro-Alarcon *et al.* (*104*) | | | ✓ | ✓ | | ✓ | | ✓ | | | |
| Ruan *et al.* (*98*) | | ✓ | | | | | ✓ | ✓ | ✓ | | |
| Navarro-Alarcon and Liu (*97*) | | ✓ | | ✓ | | ✓ | | ✓ | | | |
| Alambeigi *et al.* (*105*) | | | ✓ | ✓ | | | ✓ | ✓ | | | |
| Hu *et al.* (*106*) | | | ✓ | ✓ | | | ✓ | ✓ | | | |
| **Imitation learning** | | | | | | | | | | | |
| Schulman *et al.* (*13*) | ✓ | ✓ | | ✓ | | | ✓ | ✓ | ✓ | | |
| Lee *et al.* (*111*) | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | | | |
| Yang *et al.* (*113*) | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| Pignat and Calinon (*107*) | | ✓ | | ✓ | | ✓ | | ✓ | | | |
| Cherubini *et al.* (*114*) | | ✓ | | ✓ | | | | ✓ | | | |
| Jia *et al.* (*42*) | | ✓ | | ✓ | | | ✓ | ✓ | | | ✓ |
| **RL** | | | | | | | | | | | |
| Balaguer and Carpin (*116*) | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| Tamei *et al.* (*68*) | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| McConachie and Berenson (*38*) | ✓ | ✓ | | | | | ✓ | | ✓ | | |
| Colomé and Torras (*119*) | | ✓ | | ✓ | | | ✓ | ✓ | | | |
| Clegg *et al.* (*4*) | | ✓ | | | | | ✓ | | | ✓ | |
| Matas *et al.* (*37*) | | ✓ | | ✓ | | ✓ | | | | ✓ | |
| Petríkk and Kyrki (*33*) | | ✓ | | | | ✓ | | | | ✓ | |
| Lin *et al.* (*121*) | ✓ | ✓ | ✓ | | | | ✓ | | | ✓ | |

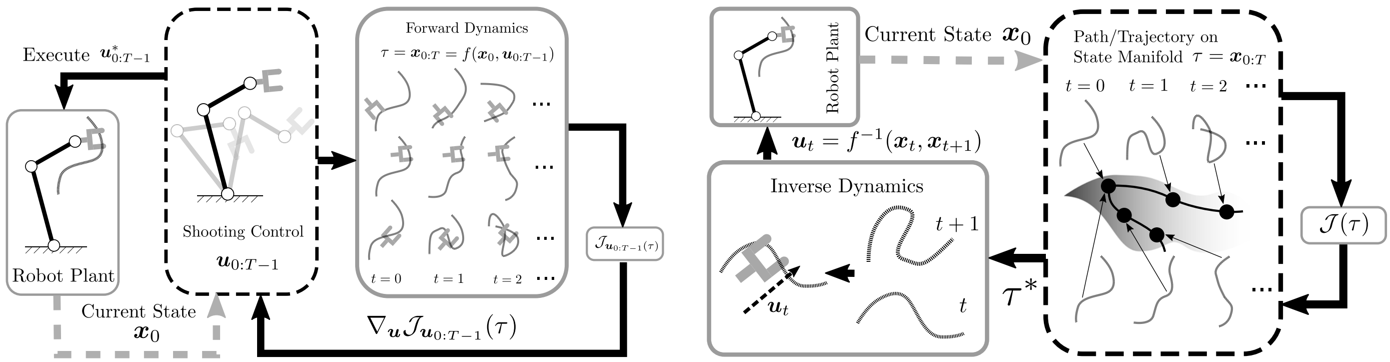


**Fig. 5. Planning manipulation of deformable objects.** (**Left**) Action shooting. The decision variables are in the action and control space (dashed rectangle). Accurate dynamical models and efficient backward gradient evaluation are the central components. (**Right**) State trajectory exploration. The decision variables are in the state space, and actions are derived from state trajectories and inverse dynamics. Efficient state representation is the key to explore dynamically feasible trajectories on the task state manifolds.

graphics works demonstrate more complex skills such as manipulating cloth (*80*) or swinging, wringing, and putting on a scarf (*27*) using ARCSim (*81*) and Dynamic Animation and Robotics Toolkits (DART) as simulator. In (*3*), a dressing assistance system is presented that uses hierarchical planning over the PhysX engine (*28*).

### *Searching state trajectories*

Planning manipulation actions can also be performed by exploration in the object state space, e.g., finding a low-cost path of $\mathbf{x}_t$, and then generating actions that cause the transition at each time step (see Fig. 5, right). The challenge here is to efficiently search or sample dynamically feasible configurations given the size of the state space. Thus, relevant work focuses on efficient definition of state representation and the transition model. (*40*) is among early efforts on developing a task-specific representation in the context of cloth folding, by focusing on poses under the gravity and bimanual grasps. Further research exploits the state space subject to high-speed dynamic for tying ropes or cloth folding in the air (*82*). For objects with simple geometry such as elastic rods (*53*), optimal conditions of equilibrium configurations may be analytically derived. The condition parameters can be deemed as an alternative state coordinate with a much smaller size and be exploited for planning of quasi-static manipulation in cable assembly (*83*). The work in (*84*) also addresses planning efficiency through a reduced state-space representation. The object state is approximated by two grasping points and a virtual band, and a model is learned to reject invalid states.

### *Learning dynamics for planning*

In data-driven approaches, a dynamical model is not defined a priori but learned, and it is usually formulated as a supervised learning problem (*85*, *86*). These approaches concern the reliability of model prediction, resorting to learning with tailored features, compact representations, and model architecture priors. An early work using neural networks predicts the 2D shape contour of a sponge or a soft ball (*85*). More recently, compact latent representations are leveraged for encoding voxelized cloth in simulation (*86*), learning terrain deformation from raw image data (*87*), performing fast simulation of several interacting objects (*88*), and visual planning (*89*–*91*). Most of the works facilitate a start and a goal image, and planning consists of generating a set of viable robot trajectories between these two. Trajectories may be generated on the basis of human demonstrations (*92*), and a plan represented in some latent (*6*, *90*, *91*) or original (*89*) state spaces. The inverse dynamical model can also be learned to infer the control actions to execute the plan (*90*).

Universal approximators such as neural models do not exploit the knowledge of deformable object dynamics explicitly. Recent machine learning works seek to embed particle-based models into the neural architecture, effectively learning a differentiable deformable simulator in an end-to-end manner (*5*, *93*, *94*). The basic idea is to adopt an interaction graph representation, in which particles are treated as graph nodes, whereas interactions among particles are modeled by edges. To reduce the complexity of a fully connected graph, the literature considers smoothing kernels to encode local interactions for fluids (*93*) or building the graph connectivity in a dynamical manner (*5*). Moreover, a hierarchical representation (*94*) is often adopted for quickly propagating long-distance effects, e.g., force transmission within rigid bodies (*5*). These models demonstrate advantageous performance in scenarios involving deformable dynamics, such as poking deformable bodies (*94*) and merging liquids (*5*). The automatic differentiation is also exploited in fluid parameter identification (*93*) and trajectory optimization, which results in an improved sampling efficiency than baseline RL algorithms (*5*, *93*).

In summary, motion planning of deformable objects is still mostly focused on various pick-and-place tasks and rather simple interactions such as poking and squeezing. An important research problem is in-hand manipulation that will also put requirements on dexterous hands, soft robot bodies, and in-hand sensing (*10*).

## Control strategies for deformable object interaction

Whereas planning an entire motion trajectory can be used when reliable nominal models are available, control-based approaches are used when this is not the case. Sensory feedback can be used to address deformable object manipulation under various uncertainties through closed-loop control. In deformable object manipulation, synthesizing a global control policy with theoretical guarantees is nontrivial because of the complex object dynamics. Thus, local controllers are used, and regulation control is performed by exerting force or velocity at fixed operating points, typically following

$$u_t^* = u(\phi(x_t) - \phi(x_g), \theta) \tag{10}$$

where the control regulates the mismatch between current and goal state $x_g$, possibly in a feature space $\phi(\cdot)$. Because of real-time requirements, $u$ is often derived using simple rules, such as a linear function between operating and feature points (*95*–*97*) or a numerical optimization process for an instantaneous step (*98*, *99*). Examples are proportional-integral-derivative controllers used for shape control of planar objects (*78*, *96*), meat cutting (*100*), and s` grasping (*101*).

More control approaches are developed on the basis of the amount of structure put into the model (see Fig. 6). Pure visual-servoing frameworks (Fig. 6, left) regulate velocities using some visual features, e.g., cloth wrinkles (*102*), to achieve the goal state. (*95*) proposes to characterize the nonrigid behavior between reference and grasping points. Intuitively, the points close to the grasped area are more "controllable" than the ones far from it (Fig. 6, middle). This approach has also been used for inserting a thread through a set of loops (*103*). More physical plausibility is accounted by addressing an inverse simulation problem. This is demonstrated in simulated cloth manipulation (*98*). (*99*) adopts a linear FEM and quasi-static assumption for manipulating 3D deformable solids, using (*22*) to estimate the material parameters offline.

The relation between control and feature points, e.g., the parameter $\boldsymbol{\theta}$ in Eq. 10, can also be estimated from the data (Fig. 6, right). Initial work in this direction considers shape deformation regulation (*97*, *104*). The goal of online estimation is to identify the mapping of end-effector displacements into the feature space. Similar ideas have been developed for manipulating 3D heterogeneous objects in surgical robotics (*105*) and performing visual-servoing tasks such as reshaping a cloth under occlusions (*106*).

## Learning-based approaches

Learning-based approaches address a variety of deformable object manipulation tasks, and the training data originate from an oracle, e.g., human expert, or robot exploration.

### *Imitation learning*

One example of acquiring manipulation skills is to provide a robot with a set of desired deformable object states, possibly with its associated control strategies that a robot may use in a slightly different setup. This is known as behavior cloning, often formulated as a supervised learning problem where the robot should imitate the observed behaviors, e.g., by synthesizing state/action trajectories that fulfill the same task goal. Desired behaviors are typically collected from human demonstrations, e.g., how to manipulate a deformable object with two hands. Imitation learning often faces unique challenges such as generalizing from a small amount of data (*107*, *108*).

Early imitation learning works focus on extracting reference motion or primitives for reproducing deformable tube plugging (*109*) or in-air knotting (*110*). The demonstrations are collected through teleoperation. Subsequent research generalizes motions for configurations that are very different from demonstrations (*13*, *108*) and connect deformable shape registration to adaptation of demonstrated trajectories. Such an approach and its variants are shown to achieve rope tying (*111*), cloth folding (*108*, *121*), and suturing (*13*). A further extension is the idea of task-parameterized models (*107*), enabling generalization of dressing movements to different human arm postures. More recent works exploit the advancement in hardware and machine learning development to collect, annotate, and extract features from a large set of demonstration data. In (*113*), an end-to-end policy is trained on 28,000 image and action pairs to form a single towel fold. (*114*) labels around 7000 images for molding kinetic sand. A neural network model is trained to predict pushing points, given current and desired shape contours of the sand. Synthetic data has been used to learn collaborative manipulation of a cloth (*42*) and tying a rope (*115*).

### *Reinforcement learning*

RL acquires a control policy through trial-and-error interactions, e.g., episodic rollouts. The learning procedure is driven by the rewards collected during the interaction. The main challenge lies in finding policies in an efficient, complete, and reliable manner, given sparse and delayed rewards. An exhaustive exploration is rarely used in practice, and works reside on using the domain knowledge such as representation or policy design or informative samples such as task demonstrations. Physics-based simulators offer a viable surrogate for the policy evaluation and update, and then the focus is on learning a robust policy that can be transferred to a real robot.

Many approaches initialize RL with human demonstrations to reduce the cost of iterating policies. This has been shown effective in dual-arm folding and flipping a clothing item (*116*, *117*). Designing appropriate task features is also vital for efficient RL to better characterize the difference between nonoptimal and optimal behaviors. This is exemplified in (*68*), where topological coordinates are used to calculate writhe as a robust description of the cloth-limb relation in a dressing scenario. As a result, RL can quickly adapt initial demonstrations for new poses in a few learning iterations. In (*118*), spherical coordinates are used to characterize the hand motion with respect to a human head model, facilitating learning dual-arm motion to put on a knit cap.



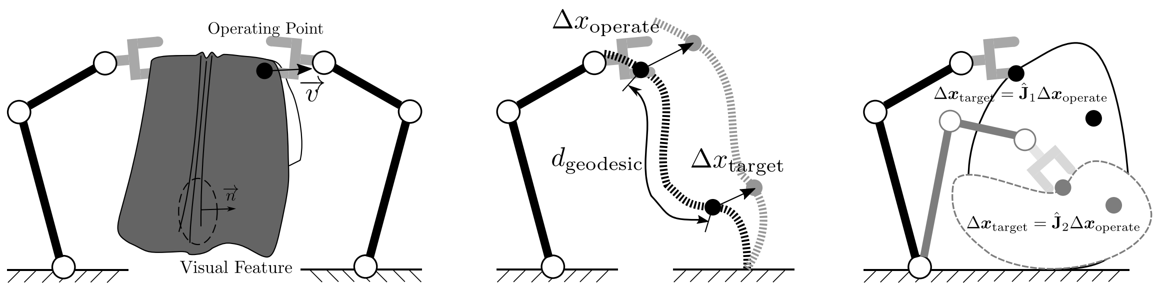


**Fig. 6. Manipulation control of deformable objects with sensory feedback.** (**Left**) Simple rules based on visual features, e.g., servoing velocity command given the orientation of surface wrinkles (*102*). (**Middle**) Approximating the relation between operating and target points with more physical insights, e.g., reduce the rigidity of Jacobian depending on geodesic distance on the deformable object $\Delta x_{\text{target}} = \alpha(d_{\text{geodesic}})\mathbf{J}\Delta x_{\text{operate}}$ (*95*, *98*, *103*). (**Right**) Estimating local linear relations based on perturbations and collected data. Controllers manage to adapt with different linear mappings, e.g., $\mathbf{J}_1$ and $\mathbf{J}_2$, in the vicinity of states before and after a large deformation (*97*, *104*).

Embedding structure to improve RL efficiency can also be made in policy design. Searching policy parameters in a restricted task-relevant space can be more efficient than using an overparameterized form such as a neural network. (*119*) iteratively extracts a linear subspace of manipulator degrees of freedom (DOFs) to search for related motion primitives while omitting irrelevant dimensions. Results show that the algorithm achieves more robust trajectory update in learning bimanual shirt folding. In (*38*), the authors use a set of approximated Jacobian models, such as the ones with diminishing rigidity (*95*) as the target policy. The models are selected in the framework of context-based multiarm bandit problem. The method demonstrates efficiency in solving rope tying and cloth placing in the Bullet simulation.

Simulators open the possibility of performing learning iterations in simulated environments. (*4*) demonstrates this in the context of animating a character to dress itself. This is achieved by a substantial reward engineering effort, hand-designed subtask composition, and an exhaustive training. The incorporation of simulation data is also beneficial for RL itself because the agent can leverage a physical model known a priori. However, when deployed in reality, policies and trajectories trained in simulation face a notable sim-to-real gap, especially for soft bodies. This raises a fundamental problem of domain adaptation. In (*37*), the authors train a deep RL policy to fold a cloth or to place the cloth on a hanger. The policy is trained in a Bullet environment that uses a kinematic constraint to emulate the grip of clothing objects. The training also benefits from many improvements of the standard policy search algorithm, as well as 20 demonstrations for each task. Domain randomization is used to make the policy robust to irrelevant conditions such as camera position and cloth size. The system achieves more than 80% success rate while around 40% for transferred skills. (*33*) approaches the strip folding problem in (*120*), by also transferring a policy learned in the MuJoCo simulator. Grid joint parameters of the strip are randomized to simulate different materials, and virtual joint forces are penalized to prevent a slip in reality. The transferred controller demonstrates an advantageous performance for materials with various stiffness.

## DISCUSSIONS AND OPEN PROBLEMS

The trend of embracing learning-based approaches in both perception and manipulation is evident by the number of works published in recent years (see Tables 2 and 5). The maturity and accessibility of deformable object simulators make it now possible to generate large amounts of relevant data (*39*, *121*) and, to some extent, compensate the need for analytical modeling. Examples include learning a robust controller from perturbed task data (*33*, *37*) or directly estimating task dynamics from interaction data (*6*). It is worth to note, however, that insights from analytical modeling are also exploited in designing the parametric form of learning models. For instance, local interactions among material particles are reflected by the adopted graph neural network (*5*, *93*) and bidirectional recurrent structures (*89*). To that end, analytical models and data-driven paradigms are becoming increasingly intertwined.

### Modeling

High-fidelity physical models, such as FEM, have enabled progress in parameter identification and state estimation (see Table 2). Mass-spring models are suitable for applications where objects are simple and real-time performance is critical. Integrated into simulators, these techniques have been used in synthesizing or learning manipulation skills (*3*, *33*, *37*, *98*).

Complex manipulation tasks, such as interaction with food items, require accurate modeling of interactions between deformable objects, manipulator, and the environment. High-fidelity simulation that involves complex contacts or large deformation is still expensive. Efficiency and robustness are also paramount for applications such as RL, where simulators can be used to collect interaction data or evaluate gradients. Despite simulators such as Bullet and MuJoCo, current manipulation works often avoid an explicit model of the friction contacts in grasp interaction and instead choose to anchor objects to end effectors with kinematic constraints (*33*, *37*). Advanced soft contacts modeling (*78*, *122*), e.g., accounting for traction and torsional frictions, can benefit robust deformable object manipulation.

Differentiable simulators are gaining attention in robotics for the efficient backward evaluation in a high-dimensional state space. Pioneering works explore this for learning-based dynamics (*5*, *93*), differentiable cloth simulation (*80*), and soft robotics (*123*). However, relevant techniques and toolkits are still in their infancy and not yet adequately exploited or validated in real complex applications.

### Perception

Estimating the state of a single deformable object, given a known model, has been addressed with the advent of fast simulators and depth sensors (*22*, *35*). Recent focus is on improving the correspondence assignment under partial views, for which structure priors are exploited (*56*). Open problems involve reliable tracking in less controlled scenarios, e.g., involving multiple objects and their interactions (*23*). Robust tracking is a prerequisite for feedback control and planning under occlusions and uncertainty. In practice, it is common to deal with objects made of different materials, such as garments composed of various textiles. However, current literature on identifying material parameters largely focuses on homogeneous materials, with the exception of (*61*). Moreover, the cost of identification remains expensive for online processing. Perception for deformable objects has witnessed successes through crafting domain features and adopting data-driven approaches, e.g., for vision-based grasping point selection (*41*, *43*, *44*). However, standard features, datasets, and benchmarks are still lacking for bootstrapping research progress and for the performance evaluation in reality.

### Manipulation

Planning in deformable object manipulation has migrated from using task-specific geometries, such as 2D shape contours (*40*), to using general physical models (*3*, *53*, *77*) thanks to the progress of modeling and simulation techniques. Because of the sim-to-real gap, a calibration of dynamic parameters is often needed (*29*, *77*). Most of the existing research focuses on motion planning, such as solving a discrete trajectory segment for forming a fold or dressing a sleeve. Integrating task planning for manipulation involving multiple stages, e.g., relying on a hierarchical action space (*3*), is not well studied yet. Recent steps toward this were made in rope reconfiguration (*89*, *91*) and cloth folding (*90*), where a compact latent space is often used for effective planning. To enable such tasks, planning grasping and regrasping will also be a prerequisite. Current literature assumes given grasps (*33*, *77*) or a geometry-based reactive strategy (*43*, *44*). This is inadequate when more complex grasps need to be planned, e.g., pinching close fingers to grasp a flattened cloth or separating a folded cloth to grasp the desired layer (*124*). Moreover, except (*125*), very few works have so far considered exploiting passive dynamics of deformable objects to generate highly dynamic motions. Dynamic motor skills can be desirable and effective in certain scenarios, such as placing a large tablecloth by swinging it in the air (*126*).

Control actions are mainly derived from an approximated Jacobian relation (*95*, *98*) to visual features (*42*) and fast inverse simulation (*99*), yielding shape control and goal state regulation (*97*). Possible extensions include servoing upon other modalities, such as contact forces (*102*). Progress can also be made by using feedback control to compensate poorly planned motion due to model mismatch. For the analysis of controllers, theoretical guarantees are usually overlooked because of the complexity of deformable dynamics, unless strong assumptions are made about the underlying model and control loops (*96*, *97*).

As in the case of rigid objects, learning for manipulating deformable objects faces difficulties of efficiently collecting and annotating data. Although the availability of simulators and self-supervision provide some viable avenues, success is only achieved for simple skills and objects, such as hanging a cloth or moving a rope (*37*, *91*). For imitation learning, demonstrating the manipulation of deformable objects often involves multiple manipulators, end effectors, and the coordination among them. More intuitive systems and techniques beyond kinesthetic teaching (*107*) need to be investigated. Most importantly, a fundamental question is representation learning. Learning and transferring with crafted domain representations have demonstrated benefits in terms of data efficiency and robustness (*68*, *69*). More general task representations and automatic retrieval of suitable representations are important and interesting for further advancing learning-based approaches.

### Systems and applications

We observe that most perception and manipulation applications focus on 2D problems (see the "Dimension" column in Table 5), especially cloth-like objects. This is probably due to the representation simplicity, e.g., a 2D polygon might suffice for many tasks (*40*, *41*). Various tasks in fashion and recycling industries, as well as assistive robotics (*3*, *68*, *69*), are important driving factors of research in this area. Rope-like objects have important roles in cable assembly (*83*) and medical suturing (*13*, *66*). Better modeling of complex deformable objects is needed for shape forming (*114*); robotic surgery (*2*); and cutting, slicing, and preparing food (*100*).

Coping with more DOFs of deformable objects solicits systems with higher manipulation dexterity, e.g., with two arms (see the "Manipulators" column in Table 5). Still, problems remain open on how to account for the coordination and cooperation, for instance, of a bimanual system (*127*). This can be useful for developing better skill representations, stress regulation, and task decomposition in deformable object manipulation. Meanwhile, it is not completely clear what types of end effectors are optimal to grasp or to nonprehensilely constrain deformable objects and whether commonly used grippers are the right choice in the first place (*124*).

Assessing different methods and systems desires common testbeds such as benchmarks including representative objects, tasks, and test data. The importance of benchmarks for identifying performance gap and driving research progress has been witnessed in the machine learning community (*31*). The recent proposals on evaluating simulated tasks (*39*, *121*), bimanual skills for cloths (*126*), and semideformable objects (*128*) make a first step toward this. Other rich and well-established benchmarks are still needed for general deformable object manipulation. It is worth noting that many of the reviewed modeling techniques and data-driven methods are equally applicable for analyzing and actuating soft robot systems. We refer interested readers to (*129*).

## CONCLUSION

The work on modeling, perception, and control of deformable objects is still in its infancy, but the large amount of recent work witnesses its importance. We identify the main challenges for deformable object manipulation in terms of the high dimensionality of the state space and the complex dynamics of deformable materials. The progress of simulation techniques has enabled various applications such as state estimation and motion planning. More importantly, the rise of learning-based approaches serves as evidence for their flexibility and effectiveness as complementary paradigms. A recent trend starts blending the two paradigms, by compensating undermodeled aspects with data or facilitating learning performance with model priors. The priors may resemble analytical models as the structure or provide task data through simulators.

## REFERENCES AND NOTES


1. P. Jiménez, Survey on model-based manipulation planning of deformable objects. *Robot. Comput. Integr. Manuf.* **28**, 154–163 (2012).
2. I. Leizea, A. Mendizabal, H. Alvarez, I. Aguinaga, D. Borro, E. Sanchez, Real-time visual tracking of deformable objects in robot-assisted surgery. *IEEE Comput. Graph. Appl.* **37**, 56–68 (2017).
3. A. Kapusta, Z. Erickson, H. M. Clever, W. Yu, C. K. Liu, G. Turk, C. C. Kemp, Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Auton. Robots* **43**, 2183–2207 (2019).
4. A. Clegg, W. Yu, J. Tan, C. K. Liu, G. Turk, Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Trans. Graph.* **37**, 179 (2018).
5. Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, A. Torralba, Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids, in *Proceedings of the International Conference on Learning Representations (ICLR)* (ICLR, 2019).
6. W. Yan, A. Vangipuram, P. Abbeel, L. Pinto, Learning predictive representations for deformable objects using contrastive estimation, in *Proceedings of the Conference on Robot Learning (CoRL)* (PMLR, 2020).
7. F. F. Khalil, P. Payeur, Dexterous robotic manipulation of deformable objects with multi-sensory feedback - A review, in *Robot Manipulators Trends and Development*, A. Jimenez, B. M. A. Hadithi, Eds. (IntechOpen, 2010), chap. 28, pp. 587–620.
8. J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, Y. Mezouar, Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey. *Int. J. Robot. Res.* **37**, 688–716 (2018).
9. F. Nadon, A. J. Valencia, P. Payeur, Multi-modal sensing and robotic manipulation of non-rigid objects: A survey. *Robotics* **7**, 74 (2018).
10. A. Billard, D. Kragic, Trends and challenges in robot manipulation. *Science* **364**, eaat8414 (2019).
11. M. Müller, J. Stam, D. James, N. Thürey, Real time physics: Class notes, in *ACM SIGGRAPH 2008 Classes*, SIGGRAPH (ACM, 2008), pp. 88:1–88:90.
12. J. Bender, M. Müller, M. Macklin, Position-based simulation methods in computer graphics, in *EUROGRAPHICS Tutorials* M. Zwicker, C. Soler, Eds. (Eurographics Association, 2017).
13. J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, P. Abbeel, A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 4111–4117.
14. Y. Kita, F. Kanehiro, T. Ueshiba, N. Kita, Clothes handling based on recognition by strategic observation, in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2011), pp. 53–58.
15. M. Macklin, M. Müller, N. Chentanez, T.-Y. Kim, Unified particle physics for real-time applications. *ACM Trans. Graph.* **33**, 153 (2014).

16. M. Macklin, M. Müller, N. Chentanez, Xpbd: Position-based simulation of compliant constrained dynamics, in *Proceedings of the 9th International Conference on Motion in Games*, MIG '16 (ACM, 2016), pp. 49–54.
17. P. Giiler, K. Pauwels, A. Pieropan, H. Kjellström, D. Kragic, Estimating the deformability of elastic materials using optical flow and position-based dynamics, in *Proceedings of IEEE International Conference on Humanoid Robots (Humanoids)* (IEEE, 2015), pp. 965–971.
18. P. Boonvisut, M. C. Çavuşoğlu, Estimation of soft tissue mechanical parameters from robotic manipulation data. *IEEE/ASME Trans. Mechatron.* **18**, 1602–1611 (2013).
19. M. Zollhöfer, M. Nießner, S. Izadi, C. Rhemann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, M. Stamminger, Real-time non-rigid reconstruction using an rgb-d camera. *ACM Trans. Graph.* **33**, 156 (2014).
20. F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, S. Cotin, SOFA: A multi-model framework for interactive physical simulation, in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Y. Payan, Ed. (Springer, 2012), vol. 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pp. 283–321.
21. E. Yoshida, K. Ayusawa, I. G. Ramirez-Alpizar, K. Harada, C. Duriez, A. Kheddar, Simulation-based optimal motion planning for deformable object, in *IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)* (IEEE, 2015), pp. 1–6.
22. A. Petit, V. Lippiello, G. A. Fontanelli, B. Siciliano, Tracking elastic deformable objects with an RGB-D sensor for a pizza chef robot. *Robot. Auton. Syst.* **88**, 187–201 (2017).
23. A. Petit, S. Cotin, V. Lippiello, B. Siciliano, Capturing deformations of interacting non-rigid objects using RGB-D data, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 491–497.
24. A. Petit, F. Ficuciello, G. A. Fontanelli, L. Villani, B. Siciliano, Using physical modeling and RGB-D registration for contact force sensing on deformable objects, in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)* (Springer, 2017), vol. 2, pp. 24–33.
25. N. Haouchine, W. Kuang, S. Cotin, M. Yip, Vision-based force feedback estimation for robot-assisted surgery using instrument-constrained biomechanical three-dimensional maps. *IEEE Robot. Autom. Lett.* **3**, 2160–2165 (2018).
26. NVIDIA, Physx sdk (2019), https://developer.nvidia.com/physx-sdk [accessed 25 September 2019].
27. Y. Bai, W. Yu, C. K. Liu, Dexterous manipulation of cloth, in *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics*, EG '16 (Eurographics Association, 2016), pp. 523–532.
28. W. Yu, A. Kapusta, J. Tan, C. C. Kemp, G. Turk, C. K. Liu, Haptic simulation for robot-assisted dressing, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 6044–6051.
29. Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, C. C. Kemp, Deep haptic model predictive control for robot-assisted dressing, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 4437–4444.
30. I. Mordatch, E. Todorov, Z. Popović, Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* **31**, 43 (2012).
31. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, OpenAI Gym. arXiv:1606.01540 [cs.LG] (5 June 2016).
32. E. Todorov, Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco, in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 6054–6061.
33. V. Petrík, V. Kyrki, Feedback-based fabric strip folding, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 773–778.
34. E. Coumans, Y. Bai, Pybullet, A Python module for physics simulation for games, robotics and machine learning (2016–2019); http://pybullet.org.
35. J. Schulman, A. Lee, J. Ho, P. Abbeel, Tracking deformable objects with point clouds, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2013), pp. 1130–1137.
36. C. Elbrechter, R. Haschke, H. Ritter, Folding paper with anthropomorphic robot hands using real-time physics-based modeling, in *Proceedings of IEEE International Conference on Humanoid Robots (Humanoids)* (IEEE, 2012), pp. 210–215.
37. J. Matas, S. James, A. J. Davison, Sim-to-real reinforcement learning for deformable object manipulation, in *Proceedings of the Conference on Robot Learning (CoRL)* (PMLR, 2018), pp. 734–743.
38. D. Mcconachie, D. Berenson, Estimating model utility for deformable object manipulation using multiarmed bandit methods. *IEEE Trans. Autom. Sci. Eng.* **15**, 967–979 (2018).
39. Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, C. C. Kemp, Assistive gym: A physics simulation framework for assistive robotics, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 10169–10176.
40. S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, P. Abbeel, A geometric approach to robotic laundry folding. *Int. J. Robot. Res.* **31**, 249–267 (2012).
41. A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrík, A. Kargakos, L. Wagner, V. Hlaváč, T. Kim, S. Malassiotis, Folding clothes autonomously: A complete pipeline. *IEEE Trans. Robot.* **32**, 1461–1478 (2016).
42. B. Jia, Z. Pan, Z. Hu, J. Pan, D. Manocha, Cloth manipulation using random-forest-based imitation learning. *IEEE Robot. Autom. Lett.* **4**, 2086–2093 (2019).
43. A. Ramisa, G. Alenyà, F. Moreno-Noguer, C. Torras, Finddd: A fast 3D descriptor to characterize textiles for robot manipulation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 824–830.
44. L. Sun, G. Aragon-Camarasa, S. Rogers, J. P. Siebert, Autonomous clothes manipulation using a hierarchical vision architecture. *IEEE Access* **6**, 76646–76662 (2018).
45. L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, L. Sabourin, Model-based strategy for grasping 3d deformable objects using a multi-fingered robotic hand. *Robot. Auton. Syst.* **95**, 196–206 (2017).
46. B. Frank, R. Schmedding, C. Stachniss, M. Teschner, W. Burgard, Learning the elasticity parameters of deformable objects with a manipulation robot, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2010), pp. 1877–1883.
47. J. Sanchez, C. M. Mateo, J. A. Corrales, B. Bouzgarrou, Y. Mezouar, Online shape estimation based on tactile sensing and deformation modeling for robot manipulation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 504–511.
48. A. Drimus, G. Kootstra, A. Bilberg, D. Kragic, Design of a flexible tactile sensor for classification of rigid and deformable objects. *Robot. Auton. Syst.* **62**, 3–15 (2014).
49. Y. Li, Y. Wang, M. Case, S.-F. Chang, P. K. Allen, Real-time pose estimation of deformable objects using a volumetric approach, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), pp. 1046–1052.
50. M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, P. Abbeel, Bringing clothing into desired configurations with limited perception, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2011), pp. 3893–3900.
51. C. Bersch, B. Pitzer, S. Kammel, Bimanual robotic cloth manipulation for laundry folding, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2011), pp. 1413–1419.
52. A. Borum, D. Matthews, T. Bretl, State estimation and tracking of deforming planar elastic rods, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 4127–4132.
53. T. Bretl, Z. McCarthy, Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *Int. J. Robot. Res.* **33**, 48–68 (2014).
54. B. Willimon, I. Walker, S. Birchfield, 3D non-rigid deformable surface estimation without feature correspondence, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2013), pp. 646–651.
55. R. A. Newcombe, D. Fox, S. M. Seitz, Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2015), pp. 343–352.
56. T. Tang, M. Tomizuka, Track deformable objects from point clouds with structure preserved registration. *Int. J. Robot. Res.* 10.1177/0278364919841431 (2019).
57. A. Tsoli, A. Argyros, Joint 3D tracking of a deformable object in interaction with a hand, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Springer, 2018), pp. 504–520.
58. J. D. Langsfeld, A. M. Kabir, K. N. Kaipa, S. K. Gupta, Integration of planning and deformation model estimation for robotic cleaning of elastically deformable objects. *IEEE Robot. Autom. Lett.* **3**, 352–359 (2018).
59. L. Bodenhagen, A. R. Fugl, A. Jordt, M. Willatzen, K. A. Andersen, M. M. Olsen, R. Koch, H. G. Petersen, N. Krüger, An adaptable robot vision system performing manipulation actions with flexible objects. *IEEE Trans. Autom. Sci. Eng.* **11**, 749–765 (2014).
60. T. M. Caldwell, D. Coleman, N. Correll, Optimal parameter identification for discrete mechanical systems with application to flexible object manipulation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), pp. 898–905.
61. B. Wang, L. Wu, K. Yin, U. Ascher, L. Liu, H. Huang, Deformation capture and modeling of soft objects. *ACM Trans. Graph.* **34**, 94 (2015).
62. P. Güler, A. Pieropan, M. Ishikawa, D. Kragic, Estimating deformability of objects using meshless shape matching, in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 5941–5948.
63. N. Bergström, C. H. Ek, D. Kragic, Y. Yamakawa, T. Senoo, M. Ishikawa, On-line learning of temporal state models for flexible objects, in *Proceedings of IEEE International Conference on Humanoid Robots (Humanoids)* (IEEE, 2012), pp. 712–718.
64. W. H. Lui, A. Saxena, Tangled: Learning to untangle ropes with rgb-d perception, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 837–844.
65. J. Grannen, P. Sundaresan, B. Thananjeyan, J. Ichnowski, A. Balakrishna, M. Hwang, V. Viswanath, M. Laskey, J. E. Gonzalez, K. Goldberg, Untangling dense knots by learning

task relevant keypoints, in *Proceedings of the Conference on Robot Learning (CoRL)* (PMLR, 2020).
66. Y. Hu, Y. Gu, J. Yang, G. Yang, Multi-stage suture detection for robot assisted anastomosis based on deep learning, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 4826-4833.
67. D. De Gregorio, R. Zanella, G. Palli, S. Pirozzi, C. Melchiorri, Integration of robotic vision and tactile sensing for wire-terminal insertion tasks. *IEEE Trans. Autom. Sci. Eng.* **16**, 585–598 (2019).
68. T. Tamei, T. Matsubara, A. Rai, T. Shibata, Reinforcement learning of clothing assistance with a dual-arm robot, in *IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2011), pp. 733–738.
69. N. Koganti, T. Tamei, K. Ikeda, T. Shibata, Bayesian nonparametric learning of cloth models for real-time state estimation. *IEEE Trans. Robot.* **33**, 916–931 (2017).
70. M. C. Gemici, A. Saxena, Learning haptic representation for manipulating deformable food objects, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), pp. 638–645.
71. G. Chance, A. Jevtić, P. Caleb-Solly, S. Dogramadzi, A quantitative analysis of dressing dynamics for robotic dressing assistance. *Front. Robot. AI* **4**, 13 (2017).
72. C. Kampouris, I. Mariolis, G. Peleka, E. Skartados, A. Kargakos, D. Triantafyllou, S. Malassiotis, Multi-sensorial and explorative recognition of garments and their material properties in unconstrained environment, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 1656–1663.
73. W. Yuan, S. Wang, S. Dong, E. Adelson, Connecting look and feel: Associating the visual and tactile properties of physical materials, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 4494–4502.
74. J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel, Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2010), pp. 2308–2315.
75. D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, K. Goldberg, Deep transfer learning of pick points on fabric for robot bed-making, in *International Symposium on Robotics Research (ISRR)* (2019).
76. L. Twardon, H. Ritter, Interaction skills for a coat-check robot: Identifying and handling the boundary components of clothes, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 3682–3688.
77. Y. Li, Y. Yue, D. Xu, E. Grinspun, P. K. Allen, Folding deformable objects using predictive simulation and trajectory optimization, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 6000–6006.
78. H. Lin, F. Guo, F. Wang, Y.-B. Jia, Picking up a soft 3D object by "feeling" the grip. *Int. J. Robot. Res.* **34**, 1361–1384 (2015).
79. I. G. Ramirez-Alpizar, M. Higashimori, M. Kaneko, C.-H. D. Tsai, I. Kao, Dynamic nonprehensile manipulation for rotating a thin deformable object: An analogy to bipedal gaits. *Trans. Robot.* **28**, 607–618 (2012).
80. J. Liang, M. Lin, V. Koltun, Differentiable cloth simulation for inverse problems, in *Advances in Neural Information Processing Systems 32* (Curran Associates Inc., 2019), pp. 771–780.
81. R. Narain, A. Samii, J. F. O'Brien, Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* **147**, 1–10 (2012).
82. Y. Yamakawa, A. Namiki, M. Ishikawa, Simple model and deformation control of a flexible rope using constant, high-speed motion of a robot arm, in *2012 IEEE International Conference on Robotics and Automation* (IEEE, 2012), pp. 2249–2254.
83. A. J. Shah, J. A. Shah, Towards manipulation planning for multiple interlinked deformable linear objects, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 3908–3915.
84. D. McConachie, T. Power, P. Mitrano, D. Berenson, Learning when to trust a dynamics model for planning in reduced state spaces. *IEEE Robot. Autom. Lett.* **5**, 3540–3547 (2020).
85. A. Cretu, P. Payeur, E. M. Petriu, Soft object deformation monitoring and learning for model-based robotic hand manipulation. *IEEE Trans. Syst. Man Cybern. B Cybern.* **42**, 740 (2012).
86. D. Tanaka, S. Arnold, K. Yamazaki, Emd net: An encode-manipulate-decode network for cloth manipulation. *IEEE Robot. Automat. Lett.* **3**, 1771–1778 (2018).
87. Z. Wang, S. Rosa, L. Xie, B. Yang, S. Wang, N. Trigoni, A. Markham, Defo-net: Learning body deformation using generative adversarial networks, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 2440–2447.
88. D. Holden, B. C. Duong, S. Datta, D. Nowrouzezahrai, Subspace neural physics: Fast data-driven interactive simulation, in *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '19 (ACM, 2019), pp. 6:1–6:12.
89. M. Yan, Y. Zhu, N. Jin, J. Bohg, Self-supervised learning of state estimation for manipulating deformable linear objects. *IEEE Robot. Autom. Lett.* **5**, 2372–2379 (2020).
90. M. Lippi, P. Poklukar, M. C. Welle, A. Varava, H. Yin, A. Marino, D. Kragic, Latent space roadmap for visual action planning of deformable and rigid object manipulation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020), pp. 5619–5626.
91. A. Wang, T. Kurutach, P. Abbeel, A. Tamar, Learning robotic manipulation through visual planning and acting, in *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22–26, 2019* (2019).
92. A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, S. Levine, Combining self-supervised learning and imitation for vision-based rope manipulation, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 2146–2153.
93. C. Schenck, D. Fox, Spnets: Differentiable fluid dynamics for deep neural networks, in *Proceedings of the Second Conference on Robot Learning (CoRL)* (PMLR, 2018), pp. 317–335.
94. D. Mrowca, C. Zhuang, E. Wang, N. Haber, L. Fei-Fei, J. B. Tenenbaum, D. L. Yamins, Flexible neural representation for physics prediction. *Adv. Neural Inform. Process. Syst.* 1806.08047 (2018).
95. D. Berenson, Manipulation of deformable objects without modeling and simulating deformation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 4525–4532.
96. J. Das, N. Sarkar, Planning and control of an internal point of a deformable object, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2010), pp. 2877–2882.
97. D. Navarro-Alarcon, Y.-H. Liu, Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-D image contours. *Trans. Robot.* **34**, 272–279 (2018).
98. M. Ruan, D. McConachie, D. Berenson, Accounting for directional rigidity and constraints in control for manipulation of deformable objects without physical simulation, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 512–519.
99. F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, C. Duriez, Fem-based deformation control for dexterous manipulation of 3d soft objects, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 4007–4013.
100. P. Long, W. Khalil, P. Martinet, Force/vision control for robotic cutting of soft materials, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2014), pp. 4716–4721.
101. M. Kaboli, K. Yao, G. Cheng, Tactile-based manipulation of deformable objects with dynamic center of mass, in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2016), pp. 752–757.
102. D. Kruse, R. J. Radke, J. T. Wen, Collaborative human-robot manipulation of highly deformable materials, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 3782–3787.
103. W. Wang, D. Berenson, D. Balkcom, An online method for tight-tolerance insertion tasks for string and rope, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 2488–2495.
104. D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y. Liu, F. Zhong, T. Zhang, P. Li, Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model. *Trans. Robot.* **32**, 429–441 (2016).
105. F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, M. Armand, A robust data-driven approach for online learning and manipulation of unmodeled 3-D heterogeneous compliant objects. *IEEE Robot. Autom. Lett.* **3**, 4140–4147 (2018).
106. Z. Hu, T. Han, P. Sun, J. Pan, D. Manocha, 3-D deformable object manipulation using deep neural networks. *IEEE Robot. Autom. Lett.* **4**, 4255–4261 (2019).
107. E. Pignat, S. Calinon, Learning adaptive dressing assistance from human demonstration. *Robot. Autonom. Syst.* **93**, 61–75 (2017).
108. J. Schulman, J. Ho, C. Lee, P. Abbeel, Learning from demonstrations through the use of non-rigid registration, in *Proceedings of the International Symposium of Robotics Research (ISRR)* (Springer, 2013), pp. 339–354.
109. M. Rambow, T. Schauß, M. Buss, S. Hirche, Autonomous manipulation of deformable objects based on teleoperated demonstrations, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2012), pp. 2809–2814.
110. S. Kudoh, T. Gomi, R. Katano, T. Tomizawa, T. Suehiro, In-air knotting of rope by a dual-arm multi-finger robot, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 6202–6207.
111. A. X. Lee, H. Lu, A. Gupta, S. Levine, P. Abbeel, Learning force-based manipulation of deformable objects from multiple demonstrations, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 177–184.
112. S. H. Huang, J. Pan, G. Mulcaire, P. Abbeel, Leveraging appearance priors in non-rigid registration, with application to manipulation of deformable objects, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 878–885.
113. P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, T. Ogata, Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robot. Autom. Lett.* **2**, 397–403 (2017).
114. A. Cherubini, J. Leitner, V. Ortenzi, P. Corke, Towards vision-based manipulation of plastic materials, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 485–490.

115. P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, K. Goldberg, Learning rope manipulation policies using dense object descriptors trained on synthetic depth data, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 9411–9418.
116. B. Balaguer, S. Carpin, Combining imitation and reinforcement learning to fold deformable planar objects, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2011), pp. 1405–1412.
117. Y. Tsurumine, Y. Cui, E. Uchibe, T. Matsubara, Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation. *Robot. Autonom. Syst.* **112**, 72–83 (2019).
118. L. Twardon, H. Ritter, Learning to put on a knit cap in a head-centric policy space. *IEEE Robot. Automat. Lett.* **3**, 764–771 (2018).
119. A. Colomé, C. Torras, Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. *Trans. Robot.* **34**, 602–615 (2018).
120. V. Petrìk, V. Smutny, P. Krsek, V. Hlaváč, Physics-based model of a rectangular garment for robotic folding, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 951–956.
121. X. Lin, Y. Wang, J. Olkin, D. Held, Softgym: Benchmarking deep reinforcement learning for deformable object manipulation, in *Proceedings of the Conference on Robot Learning (CoRL)* (PMLR, 2020).
122. I. Kao, F. Yang, Stiffness and contact mechanics for soft fingers in grasping and manipulation. *IEEE Trans. Robot. Automat.* **20**, 132–135 (2004).
123. Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, F. Durand, Difftaichi: Differentiable programming for physical simulation, in *Proceedings of the International Conference on Learning Representations (ICLR)* (ICLR, 2020).
124. J. Borràs, G. Alenyà, C. Torras, A grasping-centered analysis for cloth manipulation. *IEEE Trans. Robot.* **36**, 924–936 (2020).
125. Y. Yamakawa, A. Namiki, M. Ishikawa, Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2010), pp. 49–54.
126. I. Garcia-Camacho, M. Lippi, M. C. Welle, H. Yin, R. Antonova, A. Varava, J. Borras, C. Torras, A. Marino, G. Alenya, D. Kragic, Benchmarking bimanual cloth manipulation. *IEEE Robot. Automat. Lett.* **5**, 1111–1118 (2020).
127. C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, D. Kragic, Dual arm manipulation—A survey. *Robot. Autonom. Syst.* **60**, 1340–1353 (2012).
128. K. Chatzilygeroudis, B. Fichera, I. Lauzana, F. Bu, K. Yao, F. Khadivar, A. Billard, Benchmark for bimanual robotic manipulation of semi-deformable objects. *IEEE Robot. Automat. Lett.* **5**, 2443–2450 (2020).
129. T. G. Thuruthel, Y. Ansari, E. Falotico, C. Laschi, Control strategies for soft robotic manipulators: A survey. *Soft Robot.* **52**, 149 (2018).
130. Y. Li, C. Chen, P. K. Allen, Recognition of deformable object category and pose, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 5558–5564.
131. A. Doumanoglou, A. Kargakos, T. Kim, S. Malassiotis, Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 987–993.
132. A. C. Satici, F. Ruggiero, V. Lippiello, B. Siciliano, A coordinate-free framework for robotic pizza tossing and catching, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 3932–3939.


**Funding:** The authors acknowledge funding from the European Research Council, the Knut and Alice Wallenberg Foundation, and the Swedish Foundation for Strategic Research. **Competing interests**: The authors declare that they have no competing interests.

Submitted 17 July 2020
Accepted 30 April 2021
Published 26 May 2021
10.1126/scirobotics.abd8803

**Citation:** H. Yin, A. Varava, D. Kragic, Modeling, learning, perception, and control methods for deformable object manipulation. *Sci. Robot.* **6**, eabd8803 (2021).

# Modeling, learning, perception, and control methods for deformable object manipulation

Hang Yin, Anastasia Varava and Danica Kragic

*Sci. Robotics* **6**, eabd8803.
DOI: 10.1126/scirobotics.abd8803