Article

The International Journal of Robotics Research
2018, Vol. 37(7) 688–716

DOI: 10.1177/0278364918779698
journals.sagepub.com/home/ijr
SAGE

Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey

Jose Sanchez[1], Juan-Antonio Corrales[1], Belhassen-Chedli Bouzgarrou[1] and Youcef Mezouar[1]

Abstract

We present a survey of recent work on robot manipulation and sensing of deformable objects, a field with relevant applications in diverse industries such as medicine (e.g. surgical assistance), food handling, manufacturing, and domestic chores (e.g. folding clothes). We classify the reviewed approaches into four categories based on the type of object they manipulate. Furthermore, within this object classification, we divide the approaches based on the particular task they perform on the deformable object. Finally, we conclude this survey with a discussion of the current state-of-the-art approaches and propose future directions within the proposed classification.

Keywords

Deformable objects, robotic manipulation, cloth manipulation, deformation control

1. Introduction

Endowing robots with the ability to manipulate deformable objects spawns diverse applications with tremendous economical benefit. For instance, using robots to handle fragile products in the food industry could reduce labor cost (Buckingham et al., 2001; Tokumoto et al., 1999), manufacturing plants could use robots to manipulate flexible objects to lessen physical burden on workers (Acker and Henrich, 2003; Rambow et al., 2012), or robots could become more involved in caregiving activities (e.g. dressing, feeding) for the elderly and disabled (Chen et al., 2013; Yamazaki et al., 2014).

Although manipulation and grasping of rigid objects is a mature field in robotics, with over three decades of work, the study of deformable objects has not been as extensive in the robotics community. Another important issue is that many of the techniques and strategies devised for the manipulation of rigid objects cannot be applied directly to deformable objects. For instance, two of the main conditions applied to grasping rigid objects are *form closure* and *force closure*. The former consists of applying kinematic constraints on an object such that the object cannot perform any relative motion (Nguyen, 1988), and the latter considers a set of contact points such that contact forces can balance an arbitrary external wrench (Bicchi, 1995). When it comes to deformable objects, form closure fails to immobilize every degree of freedom of the object since deformable objects have infinite degrees of freedom (Guo et al., 2013). Similarly, force closure relies on applying a set of forces to restrain the object's movement based on its undeformed shape. However, the required contact forces applied to the deformable object rely on its deformation (Jia et al., 2011). This entails that the force closure calculation needs to be performed continuously to consider the object's new shape caused by the grasping action.

Furthermore, rigid object manipulation considers mostly the control of the grasped object's pose, as noted by Khalil et al. (2010). However, the manipulation of a deformable object also needs to address its deformation. This consideration, as well as the inapplicability of the aforementioned conditions, has led to many diverse approaches to handle deformable objects with robots.

In this survey, we review the latest advances on the topic of robotic manipulation of deformable objects. Previous surveys have covered topics such as manipulation of deformable objects in industrial applications (Henrich and Wörn, 2000; Saadat and Nan, 2002) or planning the manipulation of deformable objects (Jimenez, 2012). Henrich and

[1] Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, France

Corresponding author:
Jose Sanchez, Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, BP 10448, F-63000 Clermont-Ferrand, France.
Email: jose_manuel.sanchez_loza@etudiant.univ-bpclermont.fr

Wörn (2000) reported the first attempts of the robotic community to manipulate deformable objects. This was followed by the survey of Saadat and Nan (2002), where the objects are classified based on their shape, material, and industrial application. More recently, Jimenez (2012) presented a survey that focused on object modeling and on manipulation planning of deformable objects. Khalil and Payeur (2010) presented a very comprehensive survey, where most of the surveyed papers focus on modeling and simulation, and approaches that have been applied to industrial tasks such as assembly and food handling. However, following these surveys, the field of deformable object manipulation has seen significant contributions.

Our survey focuses on recent advances in the robotics community to address the sensing and manipulation of deformable objects, particularly in domestic and industrial applications. We propose a new classification of deformable objects that not only considers their geometric shape (as has been previously suggested by Jimenez (2012) and Saadat and Nan (2002)), but also their physical properties. We consider as deformable the objects that either (1) have *no compression strength* or (2) have a *large strain*[1] or present a *large displacement*. Objects that have no compression strength are those which do not present any resistance when two opposite endpoints are pushed towards each other. Ropes and clothes are examples of objects with no compression strength.

As noted previously, objects can also be categorized based on their geometry. Objects having one dimension significantly larger than the other two, for instance the length of a cable is much larger than its width or height, are considered *uniparametric*. *Biparametric* objects are those having one dimension considerably smaller than its other two, for instance the thickness of a sheet of paper is negligible compared with its width and height. Finally, *triparametric* objects represent solid objects. Using these criteria we classify the reviewed approaches into the following four main categories, as shown in Figure 1.

Type I: Uniparametric objects that either have no compression strength such as cables, strings, and ropes; or they have a large strain such as elastic tubes and beams. This type of object is widely referred to as **linear** in the robotics community.

Type II: Biparametric objects that present a large strain, or a large displacement, such as paper, cards, and foam sheets. In addition, thin-shell objects such as empty plastic bottles and hollow rubber balls are considered in this category. In the literature, this type of object is commonly referred to as **planar**.

Type III: Biparametric objects not possessing any compression strength. Shirts, pants, curtains, and fabric sheets are examples of this type of object. These objects are mostly known as **cloth-like**.

Type IV: Triparametric objects such as a sponges, plush toys, and food products fall in this object category. Common names used to describe these objects are **solid** or **volumetric**.

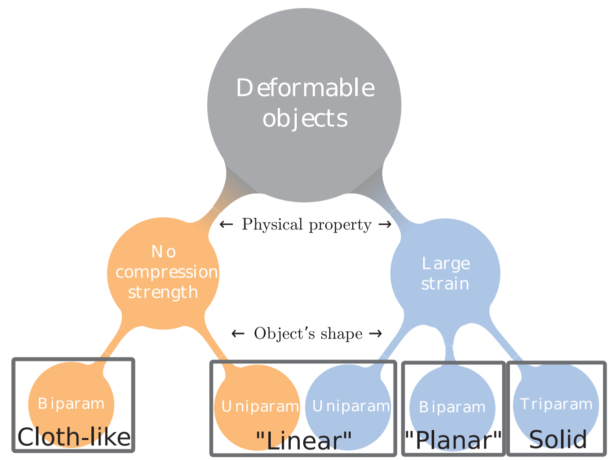

Fig. 1. Proposed classification of deformable objects.

To further divide the approaches, we group them based on their applications, and three main groups are considered: (1) sensing, (2) manipulation, and (3) task-specific. The latter group deals with tasks inherent to the object types, for instance folding a shirt or tying a knot on a rope. This proposed categorization, based on both object and application type, is succinctly presented in tables that group the approaches for each type of object.

Following this introduction, frequent concepts and terms used in the field of deformable objects are presented in Section 2. Then we review the approaches that focus on type I objects in Section 3, followed by type II objects in Section 4. Section 5 presents approaches handling type III objects and Section 6 focuses on objects of type IV. A summary of this survey and future directions are outlined in Section 7 and, finally, our concluding remarks are stated in Section 8.

2. Fundamentals of deformation modeling

This section introduces common concepts and terminology frequently used in the field of deformable object manipulation, particularly on object deformation. A deformation occurs when an external force[2] applied to an object results in the object changing its shape. Moreover, depending on the response of the object once the external force is removed, the deformation can be plastic, elastic, or elasto-plastic.

A *plastic* deformation entails a permanent deformation, that is, an object maintains the shape caused by a deforming force even when that force is removed. In contrast, an *elastic* deformation results in the object returning to its undeformed shape once the deforming force is removed (Callister, 2006). Finally, an *elasto-plastic* deformation is a combination of both elastic and plastic deformations, where the object does not return to its original shape, but it does

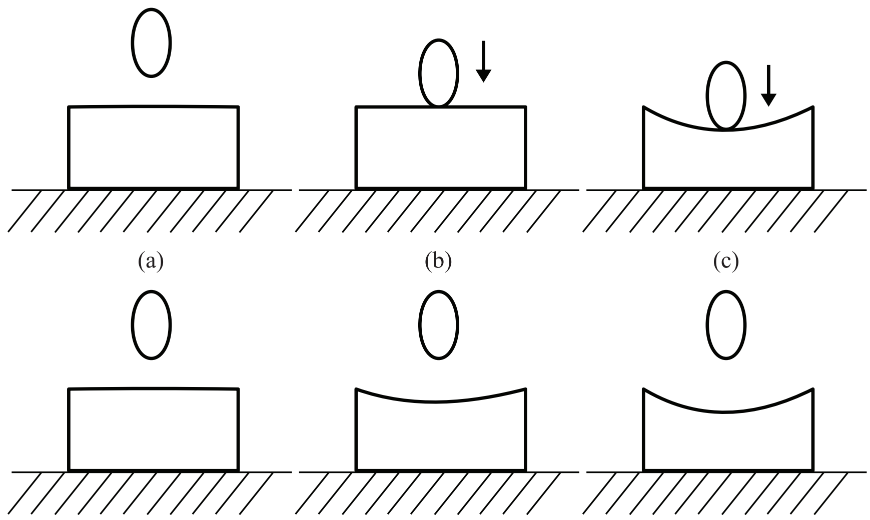

Fig. 2. Top row: a soft object being deformed by an external force: (a) before applying a force; (b) making contact with the object; (c) deforming the object. **Bottom row:** the resulting types of deformation once the external force is removed: (d) elastic; (e) elasto-plastic; (f) plastic.

not hold the deformation entirely. The deformation types are shown in Figure 2.

Figure 3 shows an object being deformed by a tensile load. The amount of deformation induced by stress is referred to as *strain* (ϵ), whereas *stress* (σ) is the ratio between the applied force F and the cross-section area A_0 (Callister, 2006). For linear elastic deformations,[3] stress and strain are related by Hooke's law (Callister, 2006):

$$\sigma = E\epsilon$$

where E is the modulus of elasticity, also called Young's modulus, and is measured in pressure units such as Pascal (N/m^2) (Askeland and Fulay, 2005).

Another important elasticity parameter is the Poisson's ratio (ν), an adimensional number that relates the ratio between axial and lateral strains (Callister, 2006). In Figure 3, the axial strain is represented by the change of length ($\frac{L-L_0}{L_0}$), where lateral strains occur perpendicularly to the applied force F.

The Young's modulus E and the Poisson's ratio ν are common parameters in modeling the deformation of an isotropic object, where the deformation's elastodynamics are represented by a set of partial differential equations solved through discretization techniques to approximate the displacement field. However, these parameters are only valid for linear elasticity. Linear deformation can refer either to a geometric or a material linearity. Geometrical linearity is not appropriate for large deformations, because only small deformations can be modeled accurately (Nealen et al., 2006). On the other hand, material linearity refers to a deformation that retains a linear stress–strain relation (Callister, 2006).

Modeling a deformation can be done with a variety of techniques. Usually, these techniques require a deformation

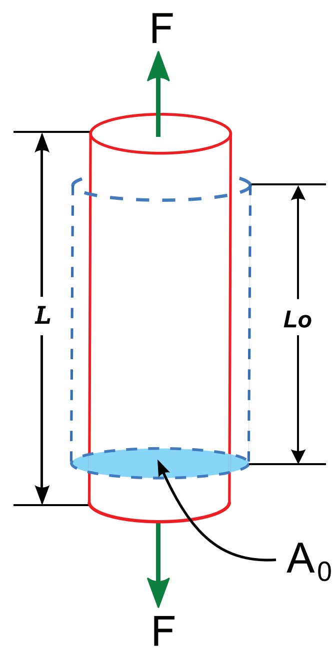

Fig. 3. A tensile load (F) producing axial and lateral strains. The blue dashed lines represent the original, undeformed shape and the red solid lines represent the deformed shape (Callister, 2006).

model and a representation of the object's shape, usually by a set of particles or a mesh. A mesh represents an object as set of points (vertices), edges, and faces or elements for a two-dimensional or a three-dimensional object, respectively. The faces are usually triangles or quadrilaterals, and the elements are commonly represented as tetrahedra or hexahedra. The deformation models provide a function to compute the position of every vertex based on their current position and an input force (Nealen et al., 2006).

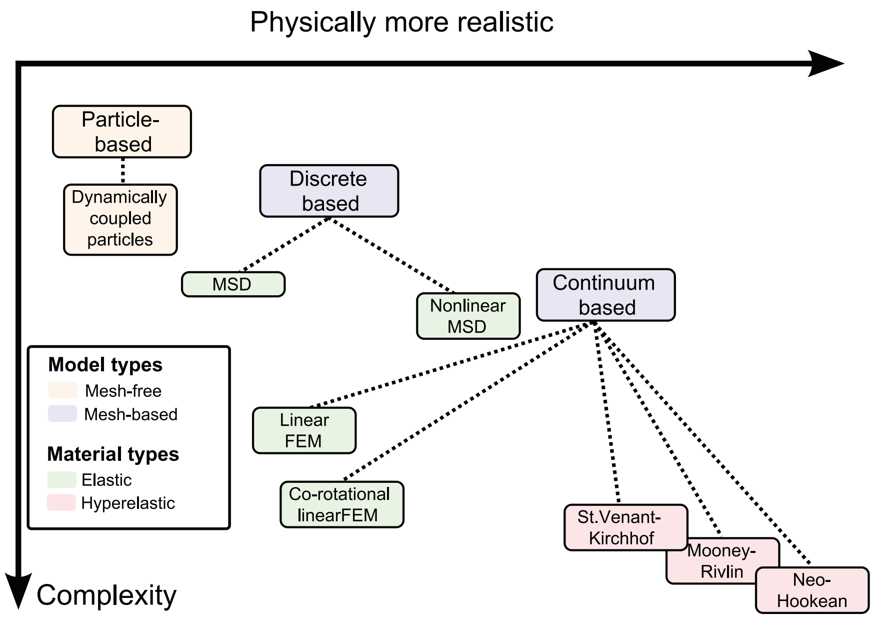

Fig. 4. Comparison of physically-based deformation models based on the evaluation results from Sin et al. (2013) and the classification presented in Nealen et al. (2006).

Deformation models[4] that do not require a mesh are termed mesh-free (or meshless), and particle-based models (Tonnesen, 1998) are an example of a meshless model. The mesh-based models are categorized either as continuum or lumped (discrete) variable models, according to the consistency of the mass and stiffness parameters with the approximated displacement fields in the elements of the mesh. The discrete-based models are mainly represented as mass–spring–damper (MSD) systems, where the vertices are treated as mass particles and the edges are considered as springs. Continuum-based methods are commonly modeled with finite-element methods (FEM), where the object is split into a set of discrete geometric parts called finite elements to approximate the object's shape (Moore and Molloy, 2007). A comparison between different physically-based models is shown in Figure 4.

MSD models are more intuitive and simpler to implement than FEM-based models, however FEM-based models are able to produce more physically realistic simulations (Moore and Molloy, 2007; Nealen et al., 2006). Furthermore, MSD models present drawbacks such as inability to preserve volume and inverting easily (Moore and Molloy, 2007; Sin et al., 2013).

This section provided merely a brief summary of interdisciplinary topics that are involved with the simulation of deformations. The interested reader is referred to Nealen et al. (2006) and Moore and Molloy (2007) for more comprehensive surveys of deformable models and modeling techniques in computer graphics. The interested reader is referred to Callister (2006) and Askeland and Fulay (2005) for a technical review of mechanical properties and elastic behavior.

3. Linear objects

The first approaches to manipulate deformable objects concentrated mostly on deformable linear objects (DLOs), also called deformable one-dimensional objects (DOOs), such as ropes, elastic rods, beams, cables, etc. One of the reasons for these initial research efforts might be the simplicity of DLOs when compared with planar or solid objects. In addition, their simulation is not as computational expensive and assumptions such as modeling them as chains of links allow for simpler algebraic solutions.

The robotic tasks related to DLOs can focus on sensing, manipulation, or a combination of both. Sensing the state of a DLO might require estimating the object's current shape, possibly caused by a deformation, or its topology (e.g. sections where an object crosses itself). Manipulation tasks of DLOs include insertion of a cable through a series of holes, untangling a rope, manipulating a tube into a desired shape, and, most commonly, tying knots, which remains the most frequently researched task in robotic manipulation of DLOs. The latest approaches dealing with DLOs by a robot manipulator are reviewed in the following sections. Some examples of the tasks involving DLOs are displayed in Figure 5, and Table 1 shows a classification of the reviewed approaches.

3.1. *Tying knots*

Tying knots tasks require the development of dedicated trajectory planning methods and force control strategies. They also require perception abilities to detect, localize, and track specific elements in ropes (e.g. intersection points) to monitor the task's progress. To meet these requirements, sensors

Table 1. Classification of approaches to manipulate DLOs using a robot. The state estimation approaches are divided into *discrete* and *continuous* depending on whether they represent the object as a finite set of points/segments or not

State estimation	**Discrete**	Lui and Saxena (2013), Matsuno et al. (2006), Caldwell et al. (2014)
	Continuous	Javdani et al. (2011), Bretl and McCarthy (2014), Borum et al. (2014)
Manipulation	**Shape control**	Rambow et al. (2012), Bretl and McCarthy (2014), Yamakawa et al. (2012)
	Untangle	Lui and Saxena (2013)
	Insertion	Weifu Wang et al. (2015), Yoshida et al. (2015)
Knot tying	**Motion primitives**	Yamakawa et al. (2008), Vinh et al. (2012), Kudoh et al. (2015)
	Algebraic formulation	Yamakawa et al. (2010), Takizawa et al. (2015)
	LfD	Lee et al. (2014), Lee et al. (2015), Huang et al. (2015)

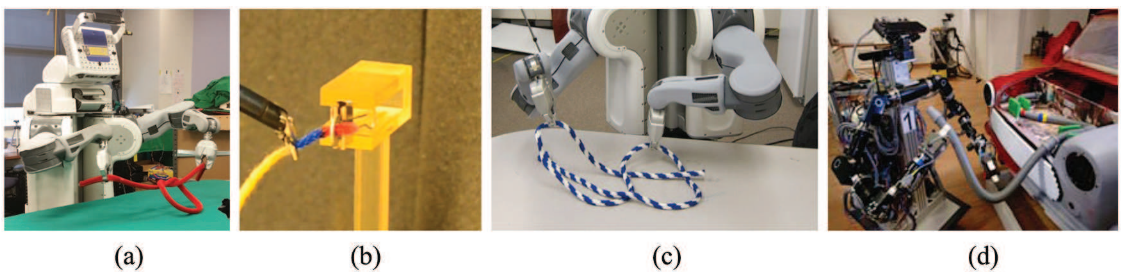

Fig. 5. Examples of different manipulation tasks performed on linear deformable objects: (a) knot tying (Huang et al., 2015); (b) string insertion (Weifu Wang et al., 2015); (c) rope untangling (Lui and Saxena, 2013); (d) shape manipulation (Rambow et al., 2012).

able to accurately measure forces, joint velocities and the state of the rope; as well as actuators able to achieve the necessary robot actions, should be carefully chosen.

Yamakawa et al. (2008) proposed a simple description of a rope to apply manipulation skills to tie a knot. This description consisted of identifying the intersections created by the rope crossing itself. Furthermore, they utilized a high-speed robotic hand with tactile sensing and a vision system to apply the manipulation skills required.

Manipulation primitives were used by Vinh et al. (2012) to perform the task of knot tying. These primitives are identified by observing how a human knots a rope using one hand. Specifically, three primitives were identified: (1) *grasping the rope*, (2) *producing a loop*, and (3) *pulling the rope* to create the knot. The robot then followed a set of points, previously extracted from trajectories demonstrated by a human teacher, to perform these primitives. However, the object's behavior is not modeled and, thus, the success of the approach depends only on a open-loop execution of these primitives by the robot. Kudoh et al. (2015) extended these manipulation primitives to tie a knot in the air with two dexterous robotic hands (i.e. the fingers are actively controlled). However, this approach is also not robust against disturbance, because no model of the object was derived and no sensor feedback was considered.

Focusing on tying a knot in the air without a physical model of the object is described in the work by Yamakawa et al. (2010). Their approach differs, however, on a dynamic manipulation of the rope that relies on high-speed sensors and actuators (1 kHz). The rope was represented as a chain of joints that is algebraically related to the robot's motion. Moreover, the position of each joint of the rope is affected by a time delay factor that is proportional to the distance from the joint to the grasping point; that is, the time delay at the grasped joint is zero, whereas the joints that are farther away get an increasing time delay. By determining this time delay parameter they were able to estimate the configuration of the rope in real time allowing the robot to tie the knot.

Most recently, learning techniques have been applied to solve the problem of tying a knot using a dual-arm robot. Specifically, in Lee et al. (2014) learning from demonstration (LfD) was used to learn a function that maps a pairs of correspondence points (i.e. from a demonstrated and a test scene), while minimizing a bending cost (warping cost). The correspondence points consist of a point cloud representing the state of the rope as a point cloud (obtained through an RGB-D sensor) and the gripper's trajectories, which were extracted from kinesthetic demonstrations. The bending cost was computed based on thin plate spline robust point matching (TPS-RPM)[5]. Also relying on the TPS-RPM to learn a warping function, the approach presented by Huang et al. (2015) used a convolutional neural network to label parts of the rope such as end-points and crossings.

Another approach, based on a geometric formulation, was proposed by Takizawa et al. (2015) where a dual-arm robot manipulated the shape of a rope to tie it around a pipe. However, their approach did not consider the tightness of the tied knot. This was shown by Lee et al. (2014) to be a necessary condition to ensure the knot stays tied to the pipe.

To solve this issue, Lee et al. (2015) extended their previous work to incorporate force information into their learning algorithm and, thus, achieve a sufficiently tight knot.

3.2. Grasping and manipulation

Several tasks that need to manipulate DLOs, such as insertion and shape control, require an accurate estimate of the object's shape as it is being deformed. This is challenging because modeling and perceiving deformation can be extremely complicated in certain configurations, e.g. when the object is deformed by an unexpected obstacle or when self-occlusion occurs. Several works attempt to overcome this issue.

Based on their previous work (Yamakawa et al., 2010), which assumes the object follows the end-effector's trajectory if the robot's motion is fast enough, Yamakawa et al. (2012) were able to deform a rope into different shapes such as a rectangular corner and a semi-circle. The inverse problem of tying ropes, namely untangling ropes, is tackled by Lui and Saxena (2013). The untangling of a rope was performed by a PR-2 robot using RGB-D data to build a graph that represents the object. To build this graph, the points obtained by the RGB-D sensor were grouped in segments by using a region-growing algorithm that reduced the representation of the rope from thousands of points to hundreds of segments. With this simpler representation, a particle filter was applied to find the best rope configuration for a set of segments. Subsequently a manipulation action was chosen so that the current configuration could be transformed into a desired configuration (i.e. an action that leads to an untangled rope).

Another desired robotic task with ropes is that of insertion, which has applications such as needle threading and assembly tasks. Weifu Wang et al. (2015) tackled the problem of inserting a rope, and a string, through a series of holes with tight-tolerance diameters with respect to the rope's diameter. To achieve this, vector fields were set in the center of each hole to drive the tip of the rope through the hole.

A similar task to inserting a DLO was presented by Yoshida et al. (2015), where an elastic band (O-ring) was extended such that it could be wrapped around a peg. Unlike, the approaches described previously in this section, this approach relied on an FEM to simulate the deformation of an O-ring. A motion planning algorithm,[6] which combined different objective functions such as collision avoidance and minimum deformation, was then used to generate a plan that inserts the O-ring into a peg with a larger diameter, thus requiring the O-ring to be deformed to insert it.

Other approaches have focused on deforming a linear object such that it reaches a desired configuration. Rambow et al. (2012) used a two-arm robot to mount a tube in a desired configuration based on a single teleoperated demonstration. The task consisted of manipulating the tube to pass by two constraining walls, thus keeping the tube in the required shape. The demonstration recorded the grippers' poses and the forces produced by the contacts with the rigid environment. As a mere repetition of the grippers trajectory does not account for the position uncertainties caused by the physically interaction with the environment, a variable admittance control was applied that allowed for a good tracking of the position when there were no contacts but decreased the tracking's accuracy to achieve a more compliant behavior, thus maintaining low contact forces while interacting with the environment. To monitor the contact forces, the robot was equipped with a force–torque sensor attached to its end-effector.

A common action among these manipulation tasks is that of reaching intermediate desired configurations to accomplish a final goal such as tying, or untying, a knot. One way to solve this is to use motion planning algorithms. The seminal work of Kavraki with Lamiraux and, subsequently with Moll (Lamiraux and Kavraki, 2001; Moll and Kavraki, 2004, 2006), focused on path planning for DLOs and planar objects, where sampling-based roadmap algorithms (e.g. rapidly exploring random tree (RRT)) were applied to find intermediate configurations of the object to build a plan that reached a specified configuration. The object was described as a parameterized curve and the sampling algorithm chose curves with minimal energy, because these were assumed to represent the most likely configuration of the object. These, and older works focusing on manipulation planning of deformable objects, have been covered in more depth by Jimenez (2012).

More recently, Roussel and Ta (2014) proposed an approach that incorporated a physics engine to their motion planning algorithm. Here, the DLO was described as a connection of nodes, where the links were modeled using an FEM, and the state space was defined by a vector of all the nodes and their positions and velocities. As the physics engine is able to determine the next state of the object based on its current state and an applied wrench, they sampled different control commands (i.e. wrenches applied by a gripper) to find the one that moved the object closer to their goal state. In addition, they exploited the environment to apply additional external wrenches to the object, and thus reach configurations that would not be possible using only a gripper. In fact, based on the test scenarios presented, allowing the object to contact the environment was necessary to solve the path planning problem. Similarly, Alvarez and Yamazaki (2016) also used a physics engine to simulate the behavior of the deformable object, in combination with a planning algorithm, while allowing a user to modify the simulated environment in an interactive manner. They showed that their planning algorithm was robust against unexpected changes to the environment. An extension of these approaches was recently described by Shah and Shah (2016), where a set of interlinked cables was manipulated into an array of clamps. This problem required the motion planning algorithm to handle an extra constraint, namely that the interlinks between cables do not become taut. They defined a series of actions for two available manipulators that were only allowed to grasp a finite set of points on each cable. Then their planning algorithm searched for the combination of actions that solved the clamping task without violating the constraint of overstretching the interlinks.

Although these approaches might be helpful to other tasks, as they solve the motion planning required to attain desired configurations, they have, so far, not been shown to work in real time as finding an adequate plan usually requires long times (e.g. minutes). Furthermore, these works have only been tested on simulation environments where the object's behavior, and its interaction with the environment, are greatly simplified.

3.3. Deformation sensing

Estimating the necessary parameters of a linear deformable object to manipulate it is a crucial issue. Several approaches have been proposed with different objectives. For some of them, the objective was to perceive the configuration of the object (e.g. are there knots or intersections and, if so, where are they located?); other works have focused on estimating the global shape by either learning parameters that describe the object's deformation or by describing the object in geometric terms. Depending on the approach, different assumptions are required as noted in the following.

Matsuno et al. (2006) were able to estimate the state of a rope using a topological description, that is, the rope was described as a node graph. The nodes of the graph represented either the ends of the rope or intersections. To distinguish whether the side of the rope was passing over, or under, they used the luminance variance on an image obtained by a stereo vision system.

A more recent approach that is able to perceive the deformation of a one-dimensional object, such as surgical suture, was presented by Javdani et al. (2011). Using a stereo vision system together with a simulator, they were able to predict the configuration of the object. Specifically, they learned a set of parameters that minimize an energy function that considers the bending, twist, and gravity effects on the object. However, this approach assumes the object does not stretch, that is, its geodesic distance remains unchanged while being deformed.

Also relying on a simulation, Caldwell et al. (2014) estimated the stiffness parameters of a flexible loop. Rather than incorporating visual information, proprioception sensing (i.e. joint values and forward kinematics) was used instead to compare with the simulated values of the object. The object was represented as a series of rigid links connected by springs, and the optimal values for the springs' stiffness were found by optimization of an error function given by the difference between measured and simulated values.

An analytic formulation to estimate the shape of a flexible tube was proposed by Bretl and McCarthy (2014). They assumed, however, that the tube cannot be stretched and is held at both ends. The object was described as a Kirchoff elastic rod, where the elastic energy was computed based on three scalar functions that measure strain. When the object is in static equilibrium, its shape can be described by geometrically solving an optimal control problem. With this formulation they were able to describe the possible equilibrium configurations of the object, based on the poses of the end-effectors holding each end, using a single global chart;[7] and, thus, estimate the configuration of the object. This theoretical framework was later evaluated in simulation and experimentally on a two-dimensional case, that is, the deformation was restricted only to remain on a plane (Borum et al., 2014).

4. Planar objects

In this section, we review approaches used on planar objects, such as printing paper, cards, and tennis balls to name a few. Most of the research, especially at the beginning of this sub-field, has focused on these types of objects as a simplified version of three-dimensional objects, e.g. in a simulated environment.

The two most frequently studied manipulation tasks on a planar object are grasping and controlling its deformation. The former consists of positioning the fingers of a robotic manipulator on the object such that the manipulator is able to lift it and hold it in the air, whereas the latter consists of changing the shape of the object into a desired configuration. Other tasks have concentrated on more specific tasks such as paper folding, manipulating the pose of the object by relying on its deformation (e.g. rotating a pizza dough-like object and reorienting a bill) and sensing the state of a deformed object. Table 2 summarizes the approaches reviewed in this section.

4.1. Manipulation

The works on manipulation of planar objects can be classified into three groups, namely, paper folding, grasping, and rotating an object. Each group raises distinct challenges due to the different types of objects they manipulate (e.g. a sheet of paper versus a thin sponge) or the specific task they focus on. Paper folding requires the robot to perform precise motions to ensure the paper is being held firmly and bent in an appropriate manner. To grasp and pick up an object, the robot must have the ability to decide where to grasp the object and how tight it should hold it. This would require different grasp forces depending on the material type, the properties of which might not be known in advance. Finally, approaches that focused on rotating objects have either exploited the environment or made use of the object's own dynamics. In the following, a review of works that address this type of manipulation is presented.

Table 2. Classification of approaches that focus on planar objects

Manipulation	**Grasp**	Guo et al. (2013), Jia et al. (2011), Gopalakrishnan and Goldberg (2004) Gopalakrishnan (2005); Jia et al. (2013, 2014)
	Pick up	Jia et al. (2014), Elbrechter et al. (2011), Kristek and Shell (2012)
	Rotation	Kristek and Shell (2012), Ramirez-Alpizar et al. (2012)
	Paper folding	Balkcom and Mason (2008), Elbrechter et al. (2012) Namiki and Yokosawa (2015)
Shape control	**Single point**	Das and Sarkar (2010), Das and Sarkar (2013)
	Multiple points	Wada et al. (1998), Hirai et al. (2001), Fanson and Patriciu (2010) Das and Sarkar (2011), Kinio and Patriciu (2012), Berenson (2013)
Sensing	**Parameter identification**	Boonvisut and Cavusoglu (2013)
	State estimation	Tian and Jia (2010), Schulman et al. (2013), Boonvisut and Cavusoglu (2014)

4.1.1. Paper folding. A clear example of paper manipulation is origami, which Balkcom and Mason (2008) addressed using a custom-made robotic system that is able to perform a sequence of folds resulting in an origami pattern (e.g. a paper plane). The system consisted on a four-degree-of-freedom arm, a suction pump to hold and move the paper, and a base with a clamp that holds the paper while the arm folds it. As no sensors were used by the system, the arm motions were performed in an open loop as selected by a simple bread-first search algorithm. As this approach relied on applying a sequence of predefined folds to the paper, no dexterous manipulation of the paper was performed.

In contrast, Elbrechter et al. (2011) presented an approach to first grasp a sheet of paper resting on a table with a bi-manual robot equipped with a five-fingered hand, and they later extended their approach to fold the paper (Elbrechter et al., 2012). Both approaches rely on a vision system with five calibrated cameras that tracked the position of a set of fiducial markers on both sides of the paper. Furthermore, they simulated the paper's deformation with a physics engine, where the paper is modeled as a two-dimensional grid of nodes. The extension of the latter approach consisted on adding a connecting link between all neighboring nodes. Each link in the physics engine was represented by a distance and a stiffness coefficient. The stiffness coefficient is decreased if a crease line, detected by the vision system, passes through any given link. Thus, the links which cross a crease present a higher deformation and can render the folding of the paper.

A similar approach, also using a bi-manual robot system, based on integrating visual information into a physics-engine simulation was presented in Namiki and Yokosawa (2015). However, the paper was modeled with an MSD model. Furthermore, a set of dynamic primitives was used to apply folds to the paper. Unlike the approaches proposed by Elbrechter et al. (2011, 2012), the paper is not covered with fiducial markers. Instead the corners are marked with different colors to allow the vision system to detect them.

4.1.2. Grasp and pick up. We consider a *grasp* as the action of holding an object through contact points, whereas a *pick up* is a subsequent action that lifts the grasped object from a supporting surface. One of the first works on grasping flat deformable objects was performed by Gopalakrishnan and Goldberg (2004); Gopalakrishnan (2005), who presented an approach that extended the concept of form closure for rigid objects to deformable planar objects. They assumed that a set of contacts, e.g. the grippers of a robot, holding a rigid object in form closure would also hold a deformable object in deform closure. Deform closure was defined as a set of contacts holding an object such that an increment of potential energy is required to release the object. Here, the potential energy is proportional to the amount that the gripper deforms the object. To determine the potential energy, an FEM model of the object and its stiffness matrix were used to compute the internal forces based on the object's deformation.

Also relying on an energy-based formulation and using an FEM model, Guo and Jia, in a series of papers, focused on grasping planar deformable objects using only two fingers. Their formulation of potential energy, which later they related to work, is based on the distance traveled by the fingers times the force they apply to the object. The translation of the fingers was described as *squeeze grasping*, where one finger moves towards the other while both maintain contact with the object. In Jia et al. (2011), they applied this approach only to hollow objects (e.g. ring-shaped objects), thus considerably reducing the computation complexity as the FEM was only simulated on segments describing the object's boundary rather than on elements describing a solid object. This was extended to solid planar objects in Guo et al. (2013) where they also defined the finger contacts as a set of points that either stick or slip, in which each of the points coincides with a node of the object's mesh. To determine whether a contact is sticking or slipping, they check whether its friction cone rotates after a change in the squeeze depth[8] (see Figure 6). Based on the state of each contact point (i.e. either sticking or slipping), they were able to infer, for instance, whether the object was slipping out of the grasp. Based on the nodes that were in contact, as well as their contact type (stick/slip), they computed a *reduced stiffness* matrix that significantly reduced the computational burden.

Later on, they introduced grasp metrics that can be applied to deformable objects, namely they defined the concept of pure and stable squeezes (Jia et al., 2013). A *pure squeeze* is defined as a grasp that generates only deformation (i.e. no translation or rotation are generated while squeezing); a *stable squeeze* is a squeeze along a vector such that it minimizes the potential energy on the object (i.e. for all the grasps with the same squeeze depth, the stable squeeze is the grasp that generates least deformation). In addition, they proposed an optimal method to resist a third finger that tries to push the object out of the grasp by moving the two grasping fingers such that the object remains grasped and the work they apply is the minimum required. Figure 6 shows an example of a squeeze grasp. It also shows a peculiarity of deformable objects, namely, the contact point grows into a contact area as the grasp progresses. Finally, Jia et al. (2014) revised the work described in Jia et al. (2011), Guo et al. (2013), and Jia et al. (2013) and presented an extensive and self-contained manuscript detailing their approach to grasp a planar deformable object using only two fingers. However, their work does not considers dynamics, nor gravity, while computing the FEM simulation. Furthermore, because the analysis is based on the assumption of linear elasticity, large deformations cannot be accurately described.

4.1.3. Rotation. Another form of manipulation that does not require the object to be grasped is that of non-prehensile manipulation. For instance, Ramirez-Alpizar et al. (2012) presented an approach that is able to rotate a deformable planar object, resting on a plate attached to a robot's end-effector, using dynamic motions consisting in rotating around and translating along a single axis of the plate as shown in Figure 7. They represented the object as an MSD model able to describe bending and compression. The elasticity parameters were estimated by experimentation. Although their approach exploits the deformation of the object to rotate it on the plate, it is not able to control the deformation, e.g. to reach a desired configuration of the object. Similarly, Kristek and Shell (2012) also proposed a method to rotate planar deformable objects using two grippers, however the applied gripper motions consisted of a set of actions that are executed in an open-loop manner as they do not use any sensor. They evaluated their approach using planar objects such as a dollar bill, a card, and various shapes of a foam material. This approach used two grippers that exploit the environment to facilitate the task of rotating the object. The first gripper controlled only one degree of freedom that slides a board so as to sweep the object against a wall to bend it enough, thus allowing the second gripper to grasp the object and subsequently rotate it.

4.2. Deformation control

This section focuses on approaches that address the problem of controlling the deformation of planar objects to achieve a desired shape. These approaches define this deformation control problem with the objective to move a target point, or a set of target points, located on the object such that once the target point reaches a specified position the overall shape of the object approximates a desired deformed shape. Thus, they indirectly control the deformation of the object through a target point, or a set of target points.

This type of problems are sometimes referred to as indirect simultaneous positioning (ISP) problems, a term proposed by Wada et al. (1998). They were one of the first researchers to address the problem of not only grasping a deformable object, but also actively controlling its deformation (Hirai et al., 2001; Wada et al., 1998). Here, the object was modeled as a mesh where the nodes are connected with springs and two sets of points (coincident with the nodes of the mesh) are defined, namely, controlled and manipulated points. The controlled points were located inside the object and were tracked by a vision system to represent the deformation of the object. The manipulated points were the contact points between the fingers of a manipulator and the object. Thus, they were able to indirectly control the deformation of the object by moving the manipulated points such that the controlled points reached a desired configuration. Recently, there have been several works attempting to solve the ISP problem. These approaches control the position of several points (multi-point control) as first proposed by Wada et al. (1998), or they focus on controlling one internal point on the object (single-point control).

4.2.1. Single-point control. Das and Sarkar (2010, 2013) used three robotic fingers to deform an object such that a single internal point on the object reached a desired position. They solved this problem using a control system that consists of a trajectory generator, a controller for each finger and the object. The finger controller is a PI controller that computes forces based on a positional error and the trajectory generator computes the desired velocities of the fingers to reach the desired position of the target point, both of which are user defined. To guarantee stability, a passivity observer monitored the energy applied by each finger controller to the object and two passivity controllers were placed at the input (trajectory generator side) and the output of the finger controller to dissipate any excess of energy. They only presented simulation results were the object's deformation was modeled using a MSD system with known coefficients.

4.2.2. Multi-point control. Fanson and Patriciu (2010) proposed a method, based on the concepts of manipulated and controlled points as described by Wada et al. (1998), that linearizes the deformation model. The model was linearized around the origin, i.e. the object's undeformed state,

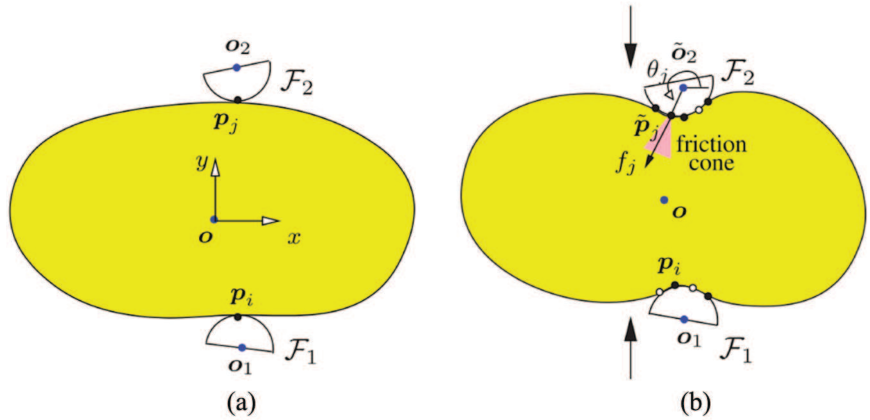

Fig. 6. A two-finger grasp (a) before and (b) after a squeeze grasp. The points represent contact points, white ones are in slip mode and black ones in stick mode (Jia et al., 2014).

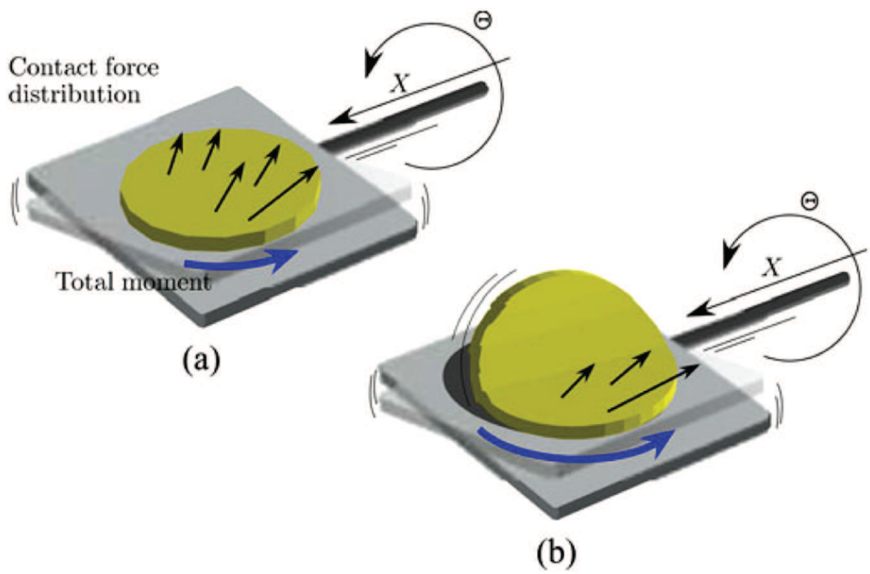

Fig. 7. The X arrow shows the translational movement and the Θ arrow the rotational motion. (Image reproduced with permission from Ramirez-Alpizar et al. (2012).)

by using one state variable that described both the position of the points along with their velocities. Having this formulation they designed an output regulator that drives the control points to a desired configuration. To compute the position of the control points based on the force applied at the manipulation points, the object was modeled as a mesh-free particle system using the reproducing kernel particle method (RKPM) (Chen et al., 1996). Although their approach was formulated to manipulate and control several points simultaneously, in their evaluation they used only a single control point and a single manipulation point. Afterwards, Kinio and Patriciu (2012) used an H_∞ controller instead and modeled the object using an FEM-based model. In their evaluations they used two control points and two manipulation points.

Das and Sarkar (2011) used an MSD model to simulate a deformable object where the links between nodes were modeled as a Kelvin–Voigt model in series with a damper. This model was used to deform a planar object such that its shape, as described by a curve, can be changed into another desired shape. This approach, however, requires a significant number of manipulation points[9] to achieve acceptable results.

Instead of relying on a model, Berenson (2013) proposed a model-free approach to manipulate an object, grasped by a set of grippers, into a desired configuration. The object was described as a set of points. In addition, it is assumed that the points of the object that are grasped by a gripper do not move with respect to that gripper. Thus, the points farther away from a gripper are affected less by the gripper's motion. This assumption, termed *diminishing rigidity*, was then used to compute an approximate Jacobian that relates the motion of the points with respect to the grippers' motions.

This approach was recently extended by (McConachie and Berenson, 2016) by choosing among a set of models for this Jacobian using a multi-arm bandit formulation. Under this formulation, the “arms" were defined as models based on their diminishing rigidity and an adaptive Jacobian (as proposed by Navarro-Alarcón et al.; see Section 6.3.2) with various sets of parameter. Although they presented simulation results on a rope and a cloth-like object this approach is better classified based on the task it performed, namely multi-point control, because the task considered by these approaches focused on reducing the overall distance between a set of target points T and a set of points P defined on the object.

4.3. Deformation sensing

These approaches focus on either determining the object’s physical parameters during a deformation or estimating the state of the deformed object. In the former case, the parameters are required as input for the object’s deformation to be simulated; whereas the latter might make use of these parameters, they can also rely solely on the output of devices such as cameras or force–torque sensors. An example of an approach to identify deformation parameters was proposed by Boonvisut and Cavusoglu (2013), where the elasticity parameters of a soft tissue phantom are estimated. To estimate the elasticity parameters, they minimized an objective function that reduce the difference between an observed deformation and an FEM simulation. The observed deformation was obtained by a stereo-vision system that tracked a set of markers on the object, whereas the inputs to the FEM simulation consisted of force and positions measurements obtained while manipulating the object.

Instead of only identifying parameters, other approaches used these parameters in physical models to track deformations of an object. One such example was reported by Tian and Jia (2010), where the deformation of thin shell objects, such as a tennis ball, is tracked by an FEM simulation that is based on shell theory (Timoshenko and Woinowsky-Krieger, 1959). Schulman et al. (2013) instead used a probabilistic model in combination with a physics-engine simulator to account for parts of an object that might not be visible to a depth range sensor. The output of the depth range sensor consists of a set of observable points which were used to infer the set of points that conform to the physical model of the object in simulation. Although the proposed algorithm was able to track objects in real time, between 20 and 50 Hz; there is no chance to recover if the algorithm diverges.

Another application on estimating the state of a deformable object was presented by Boonvisut and Cavusoglu (2014), where the constrained boundaries of an object were predicted by manipulating the object with a robot gripper. The object’s boundary was separated into three sections; namely, a fixed one, a second one that is free to move, and a third one that is manipulated by the gripper. Stereo vision, together with an FEM model, were used to track a set of points (nodes) on the object. The nodes were then grouped into patches to reduce dimensionality. With the object represented as a collection of patches, a hill climbing algorithm was applied similarly to an occupancy grid problem in order to estimate which patches are the fixed ones.

5. Cloth-like objects

In this section, we review clothes and fabric materials such as towels, sheets, and rags. Most of the approaches that deal with cloth-like objects have focused on solving one or more tasks from the pipeline described in Figure 8. The first step requires the robot to locate the clothing item, possibly identify its configuration, and decide where to grasp (e.g. a shirt’s collar). Next, the robot needs to grasp and pick up the item and, depending on the desired task, perform a preparatory action such as extending, or untangling, the garment or laying it flat on a surface. Finally, the end-task might consist on folding or hanging the item to store it or putting the item on a person (i.e. clothing assistance).

Moreover, these approaches can be grouped into three major types of tasks, namely, *sensing*, *grasp/manipulate*, and *cloth-specific* tasks. The sensing tasks are mainly concerned with recognizing the state (e.g. the shape) of the garment, or detecting ideal grasp points for further manipulation. The grasp/manipulate tasks handle issues such as picking up clothes and bringing them into a desired configuration (e.g. lay them flat on a table). Finally, cloth-specific tasks refer to tasks such as folding and hanging clothes items as well as aiding disable people with their dressing. A classification of the reviewed approaches is summarized in Table 3.

5.1. Sensing

5.1.1. State estimation. One of the most important challenges regarding sensing for cloth-like objects is self-occlusion. As clothes tend to easily crumple, key features might be covered by the object itself. We describe below how previous approaches have dealt with this issue. Kita et al. (2011) used vision sensing and simulation to recognize the shape of a garment while being held by a humanoid robot. The vision system provides three-dimensional visual data used in combination with a simulation to estimate the deformation based on the object’s size and softness. The result is a set of possible shapes defined by the point where the object is grasped. The robot reduces the uncertainty between the possible shapes by regrasping the cloth by its rim.

The configuration of different garments (e.g. sweater and shorts) is estimated using depth data acquired while a robot manipulator rotates the garment in the approach proposed by Li et al. (2014). The depth data is used to reconstruct a

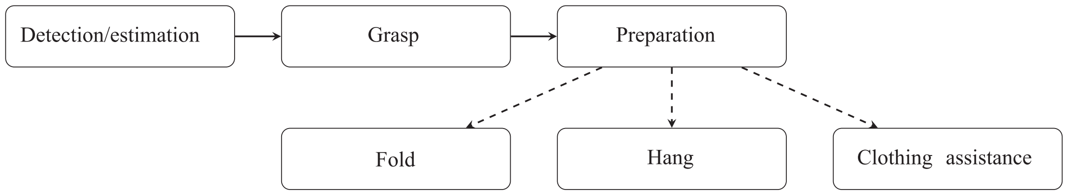

Fig. 8. Pipeline describing the common tasks of robotic cloth manipulation.

Table 3. Classification of approaches to manipulate cloth-like objects using a robot

Detection/ estimation	**State estimation**	Lee et al. (2015), Huang et al. (2015), Kita et al. (2011), Yamazaki (2014) Bergstrom et al. (2012), Li et al. (2014), Cusumano-Towner et al. (2011) Maitin-Shepard et al. (2010), Bersch et al. (2011), Twardon and Ritter (2015)
	Grasp point detection	Kita et al. (2011), Ramisa et al. (2012), Yamazaki (2014) Monsó et al. (2012), Maitin-Shepard et al. (2010), Bersch et al. (2011)
Grasp	**Grasp / lift**	Lee et al. (2015), Huang et al. (2015), Kita et al. (2011), Cusumano-Towner et al. (2011) Maitin-Shepard et al. (2010), Bersch et al. (2011), Twardon and Ritter (2015) Ramisa et al. (2012), Monsó et al. (2012), Shibata et al. (2009)
Manipulation	**Separate**	Maitin-Shepard et al. (2010), Monsó et al. (2012)
	Untangle	Lee et al. (2015), Huang et al. (2015), Cusumano-Towner et al. (2011) Maitin-Shepard et al. (2010), Bersch et al. (2011), Twardon and Ritter (2015) Doumanoglou et al. (2014); Li et al. (2015a)
	Lay / flatten	Lee et al. (2015), Huang et al. (2015), Maitin-Shepard et al. (2010) Li et al. (2015a), Sun et al. (2015), Kruse et al. (2015)
Final task	**Fold**	Lee et al. (2015), Huang et al. (2015), Maitin-Shepard et al. (2010) Bersch et al. (2011), Yamakawa et al. (2011), Miller et al. (2012) Stria et al. (2014); Balaguer and Carpin (2011); Li et al. (2015b)
	Hang	Twardon and Ritter (2015)
	Clothing assistance	Yamazaki et al. (2014), Tamei et al. (2011), Yu et al. (2017), Erickson et al. (2017)

three-dimensional mesh model of the garment. The reconstructed model is compared with the mesh models of a database to obtain the most similar one. This similarity is based on the Hamming distance between two binary vectors representing the reconstructed model and each model in the database. To construct the binary vector, a bounding cylinder is fit to the mesh model and the cylinder is subsequently divided into sections. Each section represents a binary number indicating whether or not a part of the object is in it. By concatenating the sections, a binary vector describing the three-dimensional garment is formed.

Another application for estimating the state of garment can be found in Bergstrom et al. (2012), where the focus is on estimating the folding of an article (e.g. a T-shirt and a piece of cloth). To determine the folding state of the article, they estimated its contour using a hidden Markov model (HMM) based on a single feature. This feature is extracted from a sequence of images and consists of an histogram of distance between each point on the object's contour to the rest of the contour points that is referred to as *Shape context*. This representation is much more compact and, thus, it allows for a fast estimation of the article's shape. Although a lot of local information is lost with this sparse representation, the use of temporal information proved to be more informative for correctly estimating whether or not an article was fold correctly.

In addition to the configuration, other interesting characteristics to understand the behavior of a garment are its material and fabric pattern. For instance, a laundry robot would decide either to use warm water for jeans or cool water for woolen items. This problem was tackled by Kampouris et al. (2016) using a multi-modal approach that combined RGB-D, photometric, and tactile data. They applied a variety of machine learning algorithms such as support vector machines, random forests, and HMMs, and then fused the results using a majority voting. Although their results showed that the combination of modalities was the most accurate for the majority of the test cases, there were particular cases where either using only photometric or tactile data was clearly more accurate, showing the need for a better fusion strategy.

5.1.2. Grasp point detection. The previous approaches rely on holding the garment in the air to estimate its configuration. However, before the robot holds the garment, it is necessary to grasp the garment from a possible crumpled configuration (a more likely state for clothes that have been taking out of the drier or dropped on a table). Thus, to grasp a crumpled garment a robot must first decide where to grasp.

Ramisa et al. (2012) tried to solve the issue of grasping polo shirts by detecting their collar. They used a bag of features (BoF) that is generated with scale-invariant feature transform (SIFT) features from a two-dimensional image and a histogram based on depth data. They achieved decent results where only one polo shirt was present, although when other types of garments (such as pants and T-shirts) were in the field of view of the camera, the accuracy rate reduced drastically (from ~70% to ~30%).

Also focusing on selecting grasp points, Yamazaki (2014) proposed an approach based solely on depth data to pick up a crumpled cloth. The object was represented by the edge of the cloth, along with creases created by wrinkles, as perceived by the depth data. Thus, this contour (edges and creases of the cloth) is described by points obtained from a depth sensor. Then, each point on the contour is compared to grasp points from a training dataset to select the most similar point to any grasp point in the dataset. This training dataset was generated with manually assigned grasps from a set of sample images.

5.2. Grasping and manipulation

Depending on the manipulation task and the particular state of a clothing article, the robot is required to handle the article in a particular fashion. For instance, if the article needs to be flattened or stretched to remove wrinkles, multiple regrasps might be required such that the article is held by opposite corners. The following works focused on manipulation tasks that consider different starting configurations for the articles (e.g. crumpled or lying flat).

5.2.1. Grasping for garment picking-up. A geometric approach to picking up a cloth was described by Shibata et al. (2009), where a robot generates a wrinkle by deforming the cloth so that the robot can grasp the cloth. They defined the concept of *wiping deformation*, which consists of moving the jaws of the robot's gripper towards each other while maintaining contact with the cloth. In this way, the cloth moves with respect to the surface but there is no relative motion between the cloth and the gripper at the contact points. However, this approach assumes the cloth is lying flat on a surface, a scenario that is rather uncommon since cloth-like objects tend to easily crumple.

Not relying on the assumption of the object lying flat, Monsó et al. (2012) used a robot manipulator to separate clothes. They considered the inherent sensing uncertainty to solve this problem using a partially observable Markov decision problem (POMDP). Here, the state is represented by the number of clothes on two sides of a table. The actions consist of either moving a piece of cloth from one side of the table to the other, thus reducing uncertainty, and removing a piece cloth from the table once the uncertainty is below a threshold. Picking more than one piece of cloth is considered a failure. Observation and transition models were obtained experimentally. Lastly, the goal was defined as removing all the clothes from the table with the least amount of manipulation actions.

5.2.2. Manipulation for garment reconfiguration. Cusumano-Towner et al. (2011) sought to bring a clothing article into a desired configuration using a PR2 robot. To do so, they used a HMM, a cloth simulator, and a planning algorithm. The HMM is used to estimate the garment's configuration, where the hidden state only includes the article's category and size, and the grasp points for both grippers. These grasp points represent the nodes of the garment's mesh model, according to the cloth simulator, where the gripper is grasping the garment. The robot then repeatedly regrasps the garment using the planning algorithm, which relies on the garment state output by the HMM, until the cloth is brought into a known configuration. Once the configuration of the piece of cloth is known, although not the desired one, the robot regrasps the object to bring it to the desired configuration (e.g. holding a shirt by its shoulders).

Doumanoglou et al. (2014) were able to solve this problem much faster by reducing the number of manipulation actions required to reach the desired configuration of a garment. Using a robot platform with two arms they begin by grasping the garment at a random point with one arm and let it hang to allow the other arm to grasp the lowest hanging point of the garment, e.g. at the end of a sweater's sleeve. Then, the robot releases the first grasp to leave the cloth hanging by one of its ends. Once hanging, the robot uses a POMDP to continuously rotate the garment until a desired grasp point is detected and it can be grasped by the free arm. This procedure is repeated for the second arm. Thus, allowing the robot to unfold the article by extending it by the two grasped points. To determine the two desired grasp points they used Hough forests, which are random forests using a generalized Hough transform (i.e. able to detect arbitrary shapes). To reduce the number of locations of possible grasp points, they divided the object's bounding box in an 8×8 grid.

With help of a thin-shell simulation, Li et al. (2015a) also proposed an approach to unfold clothes. They first simulate mesh models of the garments while they are grasped at different points (e.g. neck, sleeves, etc.) and let them hang loose under only the effects of gravity in a physics engine. Next, they built a database out of these models and used it to compare it against the sensed data, which consists of depth-data from multiple-views acquired by rotating the object back and forth. To reconstruct the model based on these data, they used iterative closest point (ICP) to first apply a rigid registration and then a non-rigid registration; where the rigid registration considers only translation and rotation between the sensed data and the modeled data, and the non-rigid registration accounts for the deformation of the garment. Once the model has been reconstructed they are able to grasp the required points on the garments, for instance the elbows of sweater, to unfold them.

In addition to unfolding a piece of cloth, or bringing it to a desired configuration, some approaches rely on the garment to be laying flat on a surface. Sun et al. (2015) attempted to solve this task, where a two-arm robot stretches a piece of cloth to flatten out the wrinkles. This approach uses RGB-D data to detect and locate wrinkles on the cloth and subsequently plan the required motions for the robot's arms to remove the wrinkles. The wrinkles are described as fifth-order polynomial curves and they are assigned a weight based on their volume. The largest wrinkle is then selected to be flattened by a predefined motion of the arms, which pulls the cloth to remove the wrinkle.

Using a different approach, namely hybrid force–vision control, Kruse et al. (2015) also used a two-armed manipulator to stretch a cloth. However, they include a human in the control loop that holds one edge of the cloth while the robot holds the other. This approach uses the output of an RGB-D sensor and torque sensors to maintain the cloth taut. They used a vision controller relying on the normals of each point on the cloth to compute a velocity vector. The magnitude of this velocity is based on the amount of normals that are within a user-defined threshold, e.g. the normals should point upwards to keep the cloth stretched, and the direction is chosen as the mode of the normals that are not within the threshold. The force information, obtained through the estimation of joint torques, is used by a force controller with a damping coefficient to generate a velocity that maintains a desired tension for the cloth. To obtain the velocity of the hybrid controller, both of these velocities are combined by a simple addition.

5.3. *Cloth-specific tasks*

As mentioned previously, works on cloth-specific tasks have focused on folding, hanging clothes or assisting in dressing people. To hang clothes, or put them on people, the robot is required to identify and track openings on the clothes so they can be put through a hanger or onto someone's legs, arms, or head; folding might not require any feedback as planned actions can be executed with open-loop motions.

5.3.1. Folding. One of the most popular tasks regarding cloth manipulation is that of folding. A fold can be performed either on a supporting surface, as depicted in Figure 9, or in the air as shown in Figure 10. The approaches that have tackled the problem of robotic cloth-folding have relied on a variety of techniques such as detecting grasp points, LfD, geometric representation (e.g. describing clothes as polygons), etc.

Maitin-Shepard et al. (2010) were the first to enable a robot manipulator to solve the complete task of folding clothes starting from an unorganized pile of clothes. The main focus of their work concentrated on detecting grasp points to facilitate the folding, because predefined manipulation actions execute the folding maneuver. Their algorithm assumes the cloth is of rectangular shape (e.g. a towel) and they used stereo correspondence to detect the garment's corners. Specifically, they applied random sample consensus (RANSAC) to a border classification algorithm that detects the edge of the garment based on depth discontinuities.

Another approach that focuses on detecting grasp points to fold a piece of cloth can be found in Bersch et al. (2011). Unlike in Maitin-Shepard et al. (2010), where only towels are considered due to their rectangular shape, this work enables a robot to fold a T-shirt. However, the approach relies on the T-shirt being completely covered with fiducial markers. These markers are used to infer the position where a robot's gripper is holding the T-shirt by rotating it to generate a three-dimensional model that includes a point cloud representation along with the markers' positions. Once the robot knows the location of the grasp it uses its free arm to grasp one of the predefined key points, in this case the shoulders of the T-shirt. This sequence is repeated until the robot is holding the T-shirt by both shoulders and then it executes a motion to fold the T-shirt in an open-loop manner.

In addition to detecting grasp points to subsequently fold a garment, other approaches assume this step has been performed and focus only on the folding action. For instance, Yamakawa et al. (2011) proposed an approach where two robotic hands holding a garment by its corners fold it in midair. The garment was modeled as grid of nodes, where the nodes that are grasped by the robotic grippers are assumed to have the same position as the grippers (e.g. if the hand moves, the grasped node follows the same trajectory); and the position of the remaining nodes is determined proportionally based on the distance to the grasp nodes. This position is then controlled by a single parameter, termed *time delay*, which serves as a substitute for the dynamic and elasticity parameters (mass, inertia, Young's modulus, etc.) of the object. As this formulation is an algebraic description, the approach allows to estimate the shape of the object based on the robot's kinematics. Thus, by solving a trajectory of points describing a fold of the garment, they were able to compute the corresponding joint values of the robot required to execute that fold.

Similar to the approach of Yamakawa et al. (2011), other approaches have relied on a simpler model to represent the state of a cloth, namely, a polygonal description. This polygonal description represents the surface of a flatten cloth with straight lines, i.e. reducing the cloth to a two-dimensional model. Based on this geometric modeling, Miller et al. (2012) used folding actions based on gravity (g-folds) to fold different types of garments. Two different types of g-folds were proposed, where one folds the whole stack[10] and the other folds only the previously folded part by not releasing the grasped points (e.g. after folding a long-sleeve by pivoting on the shoulder, the sleeve is then folded such that the end of the sleeve is aligned with the bottom of the

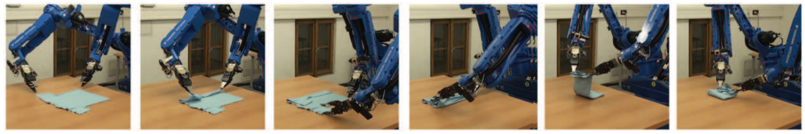

Fig. 9. A geometry-based approach to fold a clothing item laying on a table (Stria et al., 2014).

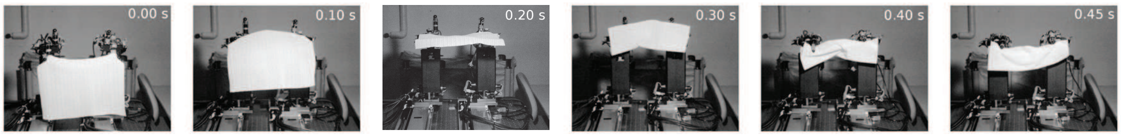

Fig. 10. Sequences of a dynamic in-air fold executed in less than half a second (Bergstrom et al., 2012).

article). Also relying on a polygonal description of a clothing article, the work presented by Stria et al. (2014) was able to fold clothes much faster, requiring seconds instead of minutes (as in Miller et al., 2012). This major speed-up is achieved during the perception phase and is due to the novel model they propose. This polygonal model uses the vertices on the clothing article to define angles at the vertices and also the lengths between vertices. Thus, having a compact description of the clothing article as a set of angles and lengths, which they encode with probability distributions for a collection of articles of the same class (e.g. pants would be a class and a short-sleeved shirt another). Both of these approaches require however that only a single garment is resting completely spread on a supporting surface. Recently, Li et al. (2015b) proposed an approach for a bi-manual robot to fold different garments such as shirts, pants, and towels. Their approach first detects key points for the robot to grasp (e.g. collar, sleeve ends) and then finds the optimal folding trajectory of the robot's arms. The trajectories are parameterized as Bezier curves, and their points are found by minimizing the cost between the desired fold and a simulated fold which is computed using a physics engine.

Machine learning approaches have also been utilized to solve the task of folding a garment using a robot. For instance, Balaguer and Carpin (2011) applied reinforcement learning (RL) to fold a towel using a two-armed robot. To reduce the search space, demonstrations were taught to the robot by a human teacher. The RL problem was formulated as finding a map between the trajectory of both end-effectors and a set of points describing the object. This set of points was obtained through eight cameras tracking the position of 28 markers attached to the object's boundary. Other approaches that rely on learning techniques to fold towels can be found in Lee et al. (2014) and Huang et al. (2015), as reviewed in Section 3.

5.3.2. Others (clothing assistance, hanging) Other applications for clothing manipulation found in the literature concern clothing assistance and hanging of clothes. One case of clothing assistance, such as helping a person put on a shirt, was tackled in Tamei et al. (2011). Here, RL was used on a two-armed robot to put a shirt on a mannequin. To reduce the task's dimensionality, they define topological coordinates,[11] on both the shirt and the mannequin; and only control one joint on each arm. A motion capturing system tracked relevant points on the mannequin and the T-shirt, which later were used to compute the topological coordinates. They start by teaching the robot how to solve the task through kinesthetic training to extract an initial trajectory, then RL modifies the way-points of the trajectory to achieve the task.

Another case of clothing assistance can be seen in the work of Yamazaki et al. (2014), where a humanoid robot helps a person putting on a pair of pants. This approach computes an optical flow on images to detect the state of the task by comparing the flow with a previously generated database of flows, e.g. it can identify whether a leg is being introduced in an opening of a pair of pants. It also incorporates depth data to estimate the location of the person's legs and force sensing is used to detect a possible failure such as having a leg stuck in the pants. If a failure is detected, the approach is able to recover by modifying a predefined trajectory. One recent approach, proposed by Yu et al. (2017), focused on predicting the outcome of dressing a person with a robot using HMMs. The outcomes, for a task of pulling a medical gown onto a person's arm, were classified as either successful dressing, missing the arm or getting the arm caught by the gown. Owing to the complexity in gathering haptic data that involves a robot to generate multiple contacts with persons and as this process is unsafe and time consuming, the authors proposed the use of a physics simulation to generate simulated haptic data. By relying on simulation, they were able to reduce risk, since the number of data gathered from real interactions with people was kept at a minimum while still achieving high-accuracy results. Estimating the forces exerted by a garment on a person, is an important issue to consider when using robots to assist people with dressing. Erickson et al. (2017) recently addressed this problem by using a recurrent neural network (RNN), a type of network architecture that is ideal for sequential data, to map variables at the end-effector of a robot to a force distribution on a person's simulated limb. The input consists of nine variables describing the force, torque and velocity measurements of the end-effector that is grasping the garment being pulled into a simulated limb; the output is a force map with hundreds of predefined points that have being uniformly distributed along the simulated limb. This approach was able to obtain visually similar results to the ground truth, and the authors proposed to extend it to real robots in the future to cope with previously unseen scenarios (e.g. higher forces and velocities not available in the simulated data).

As mentioned previously, hanging garments is another task that requires the manipulation of clothes. An application of this task is showcased by Twardon and Ritter (2015), where two robotic hands work in combination to hang a knit cap on a coat rack. The approach proposed by Twardon and Ritter begins by generating a graph representation of the cap from depth data. To generate the graph, it first processed a point cloud to detect edges based on the points' normals (e.g. by removing smooth curves). Once the edges were found, a thinning algorithm was applied to reveal the intersections of the edges. These intersections were selected as nodes if they had at least three neighbors. Finally, it used this graph to compute simple cycles to subsequently extract boundaries that represent, in this case, the opening of the cap.

6. Solid objects

Solid objects such as sponges, plush toys, and food, remain the least researched type of deformable objects. Perhaps the main reason for this are the computational costs imposed when simulating three-dimensional objects which are much higher than those of simulating planar, or even cloth-like, objects. However, recent advances in computing have increased the processing power significantly such that real time simulation of realistic deformation (e.g. using FEM) of solid objects is now possible. This progress has led to an increment of research of solid deformable objects in the robotics community, as it is shown in Table 4. Some examples of manipulating solid deformable objects can be seen in Figure 11.

Another kind of solid deformable object is soft tissue. It has been studied extensively for medical robotic applications. As medical applications are subject to constraints, not shared by domestic and industrial applications, such as dealing with moving organs in real time and limited space when performing minimally invasive surgeries; we cover only those works where the proposed methodologies could be applied to task in industrial (e.g. cutting meat) or domestic settings. For a review on medical robots and their applications to surgery, the reader is referred to the survey by Taylor et al. (2016). The manipulation tasks that have been researched for solid objects can be grouped into three main categories: *grasp/manipulate*, *deformation sensing*, and *deformation control*.

6.1. Grasping and manipulation

Similar to deformable planar objects, grasping and picking up a solid object was one of the first tasks studied in this sub-field and it remains an active research topic. Other manipulation tasks have been investigated for solid objects as well; for instance, cutting meat, splitting food items, and handling soft tissue to assist in surgery. Depending on the task, the robot might be required to manipulate the object in distinct ways, such as holding it firmly or deforming it to reach a desired configuration. Approaches focusing on the latter are thoroughly reviewed in Section 6.3.

One of the first works regarding manipulation of solid deformable objects was proposed by Howard and Bekey (1999), which consisted of estimating the minimum force required to lift an object. Here, the object was modeled as a lattice of interconnected nodes, where the connections were described by a Kelvin–Voigt model, i.e. a spring in parallel with a damper. Two planar end-effectors, equipped with force sensors, pushed the object against each other to lift the object. If the object was not lifted, or if it slipped from the grasp,[12] the force was increased until a required lifting force was found. This lifting force was stored in a database on which they trained a neural network that computed the force required to lift a new object based on its stiffness and damping coefficients. These coefficients were computed a priori by grasping the object with the manipulators and measuring the relation between the applied force and the manipulators' displacement.

More recently, Delgado et al. (2015) used a multi-fingered robot hand equipped with tactile sensors to grasp and hold a deformable object. A linear relationship is computed between the distances of the opposing fingers and the measured forces from the tactile sensors, which they termed *deformability ratio*. This deformability ratio is then used to compute the maximum force allowed to be exerted on the object, thus reducing the deformation of the object. This approach was later extended in Delgado et al. (2016) by coupling it with a grasp planner to perform tactile servoing, which consists of moving each finger until a desired force and a desired contact location on each sensor are reached.

In contrast to these approaches, other approaches require a physically accurate model[13] to grasp a deformable object. For instance, Lin et al. (2014, 2015) proposed an approach where a deformable object is grasped and lifted by a robotic gripper that relies only on a mesh model of the object and the positions of each finger, thus no exteroceptive sensors (such as tactile or force sensors) are required. To do so, they used an FEM formulation to solve the deformation simulation based on the displacements caused by the fingers. Each finger contact is modeled as a set of points that are either sliding or sticking. The approach consists of moving the fingers towards each other to squeeze the object and subsequently lift it, after a *lift test* is passed. This lift test is based on the simulated deformation generated by the set of contact points of each finger. Using this simulated deformation, they checked whether the majority of the contact points are not sliding (i.e. are sticking) to determine whether the lift test is passed. Similarly, Zaidi et al. (2017) applied an FEM simulation with an MSD model to estimate the object's

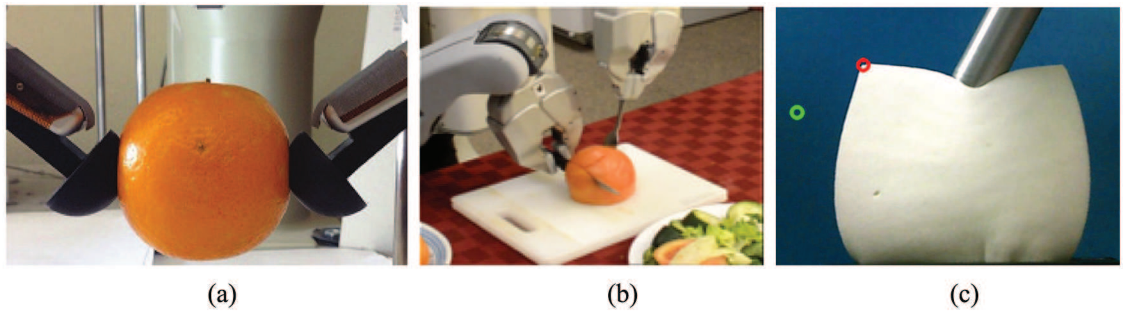

Fig. 11. Examples of different manipulation tasks performed on solid deformable objects: (a) grasp and lift (Lin et al., 2015); (b) cutting with a knife (Gemici and Saxena, 2014); (c) shape control (Navarro-Alarcon et al., 2013b).

Table 4. Classification of approaches to manipulate solid objects using a robot

Sensing	**Parameter identification**	Frank et al. (2010), Frank et al. (2014), Fugl et al. (2012), Güler et al. (2015)
	Shape estimation	Khalil et al. (2010), Cretu et al. (2010a), Cretu et al. (2010b), Cretu et al. (2012), Leizea et al. (2014) Petit et al. (2015), dos Santos et al. (2014), Malti et al. (2011)
Manipulation	**Pick up**	Howard and Bekey (1999), Delgado et al. (2015), Delgado et al. (2016) Lin et al. (2014), Lin et al. (2015), Zaidi et al. (2017)
	Split / cut	Long et al. (2014b), Long et al. (2014a), Gemici and Saxena (2014), Hu et al. (2012)
	Suturing	Iyer et al. (2013), Leonard et al. (2014a), Leonard et al. (2014b), Jackson et al. (2016)
	Shape control	Higashimori et al. (2010), Yoshimoto et al. (2011), Smolen and Patriciu (2009) Navarro-Alarcon et al. (2013a), Navarro-Alarcon et al. (2013b), Navarro-Alarcon et al. (2014), Navarro-Alarcon et al. (2016), Navarro-Alarcon and Liu (2014), Navarro-Alarcon and Liu (2017)

deformation caused by a three-finger grasp. As opposed to Lin et al. (2014, 2015), this approach computed the contact forces using a spring–damper contact model based on the fingers' positions and velocities. The computed forces are the input to the FEM simulation which outputs the new positions of the mesh's nodes, based on the MSD model, to describe the object's deformation. This work was validated using a dexterous robotic hand that grasped two soft objects, one made of rubber and the other made of foam.

Other research works have focused on more complex manipulation tasks such as cutting deformable objects. Hu et al. (2012) presented an approach to debone a chicken using a cutting robot, where cutting through ligaments, tendons, and skin is required. The cutting robot had only two degrees of freedom and followed a path plan that was derived based on key points, obtained from a stereo-vision system, of the chicken such that they form a cutting plane. As cutting through the bone is undesired due to health hazard, the robot was also equipped with a force–torque sensor to detect contact with the bone. In case of contact, the robot switched to force control and modified, in real time, the initial path to maintain the force within a specified threshold. Other efforts by the same group have focused on cutting other food products, such as potatoes (Zhou et al., 2006a,b). By using a cutting robot arm with seven degrees of freedom, Long et al. (2014b) were able to follow a curve (as opposed to a straight line as presented in Hu et al. (2012), to separate a beef muscle. Furthermore, they made use of a second robot that was rigidly attached to the meat, with a special gripper, to pull the meat to reduce the friction for the cutting robot. The pulling robot used impedance control to maintain a desired pulling force, whereas the cutting robot used position control to follow the cutting path with a visual feedback that updated changes caused by the meat deformation. However, the cutting robot was not able to follow the path completely due to its inability to overcome the required friction to cut the meat. This was later addressed by Long et al. (2014a) by adding a force–torque sensor to the cutting robot. The force control applied a slicing motion that reduced the required cutting force. Note that the meat's deformation was not explicitly modeled in these works.

Gemici and Saxena (2014) used a PR2 robot to prepare a salad, in which the robot actions required cutting and splitting food objects such as bread, cheese, and lettuce. The approach used haptic data, such as force and tactile data, to learn the objects' properties (e.g. brittleness and elasticity) by performing manipulation actions that stretch, bend and cut the objects. The properties were assigned a value from zero to one by human experts, which the robot then learned using a regression model. The different learned properties were used to classify the objects and decide a proper action for a given task. For instance, the task of cutting a cheese depends on its properties: as cream cheese is much more brittle than Cheddar cheese, the robot is required to hold the cream cheese with a fork, instead of using its gripper, so as to not damage the cheese.

Although the works presented in the following paragraphs do not directly manipulate deformable objects, as they are concerned with automated surgical suturing, they highlight intrinsic problems arising when interacting with soft tissue. For instance, inserting a needle into soft tissue causes deformation and thus, it might be necessary to either estimate the contact forces or track the deformations to adapt the suturing trajectory to reduce damage to the tissue. As suturing is one of the most common surgical tasks, automating it would result in considerable reduction times as well as in a consistency increment. Robotically assisted surgeries (RAS) have not been, so far, truly autonomous as a surgeon is still required during the procedure. The following works have proposed approaches that attempt to automate the suturing process.

Iyer et al. (2013) required only a set of entry and exit points defined by a surgeon. They used a six-degree-of-freedom manipulator with a suturing tool attached to its end-effector to give an extra degree of freedom of rotation. This tool held a circular needle to go through the set of points to suture the incision. Using a monocular camera, they estimated the location of the needle by fitting an ellipse. Once the needle was located, they computed an ideal center point for the needle to go in and out of the entry and exit points, respectively. Their results were fairly consistent, although the approach was highly sensitive to poor illumination conditions.

A further step towards full automation was achieved by Leonard et al. (2014b), as they proposed an approach capable of automatically determine the suturing points given an incision contour. They used the Smart Tissue Anastomosis Robot (SMART), which is a robotic system developed for suturing equipped with a seven-degree-of-freedom arm, a monocular camera, and a custom-made tool that rotates a circular needle to pass a thread through soft tissue. Based on the incision contour, the stitches locations required to perform the suture where computed such that they would be evenly distributed. Once the locations where the needle was required to pass were known, a tracking algorithm (based on Kanade–Lucas–Tomasi) was used to deal with deformations of the soft tissue caused by the needle. The features used for the tracking algorithm were based on the distance to the closest pixel in the target contour, the color of the foreground and a bounding box of the incision. To perform the suturing, the robot first tied a knot on one edge of the incision and then continued to the next stitch location. In addition, force sensing was used to tighten the knot without exceeding a threshold that was experimentally determined to avoid damaging the tissue. Their results were four times faster than surgeons using the same custom-made tool and they also achieved higher accuracy. They later extended their approach in Leonard et al. (2014a) by adding a multi-spectral vision system to detect near-infrared fluorescent (NIRF) markers. The addition of NIRF markers allowed their system to deal with occlusions, because they can be detected through tissue and blood. However, as their approaches relied on a homography, they were limited to dealing with only flat surfaces on the phantom.

Focusing on a different aspect of automated suturing, Jackson et al. (2016) presented an approach to estimate the deformation forces of the tissue caused by the needle. Since tissues can be damaged by excessive force, their approach aims to minimize the internal forces produced by the needling while suturing. Instead of using a vision system, their approach relied solely on position and force sensing, in combination with an unscented Kalman filter (UKF), to estimate the forces caused by the needle. Although the UKF produce accurate results, it was not run online in their experimental validation.

6.2. *Deformation sensing*

The approaches presented in this section address either *parameter identification*, to later use those parameters in a deformation model, or *shape estimation*, where sensors are used to directly measure the deformation (or shape) of the object without the need of identifying parameters, as they are known in advance or not required at all. Although the former approaches require to actively deform the object to obtain sensor measurements (e.g. through force/tactile or vision sensors) to infer certain deformation parameters; the latter approaches mostly focus on detecting changes on the object's shape as it is being deformed by another agent (e.g. a surgeon manipulating soft tissue) and using those changes to provide a three-dimensional visualization of the deformed object.

6.2.1. Parameter identification. Parameters (either physically meaningful such as the Young's modulus or task-specific, as a coefficient of deformation) can be estimated through sensor measurements with or without a model. An approach that requires a model was presented by Frank et al. (2010, 2014), where the Young's modulus and the Poisson ratio of different flexible objects were estimated. The estimation was based on force sensor measurements, obtained by probing the objects, a volumetric model of the object that was generated using depth data from multiple views and a co-rotational FEM model. Having the simulation of the deformation from the FEM model and the observed deformation, a search for the best values of these parameters was performed such that the error between the observed and simulated deformation was minimized.

Instead of an FEM model, Fugl et al. (2012) used a more limited deformation model, namely the Euler–Bernoulli beam model, to estimate the Young's modulus of a flexible object to describe its deformation. In this approach, the object was divided into sections that have a specific curvature produced by the deformation. These curvatures, along with the object's pose obtained through RGB-D data, described the deformation state of the object. As well as the previously mentioned approaches, this approach also minimized a function that considers both the sensed error and the simulated error.

Other types of models that do not require a mesh have also being explored in the field of robotics. For instance, Güler et al. (2015) used a position-based[14] physics simulation called meshless shape matching (MSM). This simulation only requires position information, which is obtained using an optical flow algorithm on a sequence of images. As the images are obtained from a single view, only one face of the object was considered for estimating the deformation. Instead of relying on the usual elasticity parameters (e.g. Young's modulus and Poisson's ratio), their approach is based on the computation of a *deformability* parameter, used in the simulation, that minimizes the error between the sensed positions and the simulated positions.

6.2.2. Shape estimation/tracking. One of the first works on deformation estimation used a stereo-vision system to track the surface of an object being deformed by a robotic gripper (Khalil et al., 2010). It integrated the estimation in GraspIt! (Miller and Allen, 2004) to have a real-time representation of the deformation. Cretu et al. (2010a) were able to track the contour of an object using a growing neural gas network while the object was deformed by a robotic hand. To track the contour, they first identified the foreground from the background using the HSV value of each pixel and later applied a Sobel filter to detect the object's contour. Lastly, the neural gas network was used to determine the minimum amount of points that described the contour. Later on, in Cretu et al. (2010b), force measurements obtained with strain gauges at the fingertips of the robotic hand, along with their positions, were used to learn the elastic behavior with a neural network that mapped the deformed contour, obtained through visual data, to the force and position measurements. Thus, the neural network input consisted of a six-dimensional vector containing the force and position information and its output consisted of the two-dimensional position of a set of points representing the object's contour. A further improvement was then made in Cretu et al. (2012), by also tracking lines that formed a grid on the object. These lines were however artificially marked on the object.

Similarly, an approach to track deformation of objects was presented in Leizea et al. (2014), with a focus on augmented reality applications. The approach consisted as well in incorporating depth data into a physical model. However, this approach used a mass–spring model to estimate the deformations caused by an external force. The chosen model resulted in a much faster simulation, between 80 and 120 Hz, with the drawback of not being able to handle large deformations. Recently, a real-time simulation using an FEM model was presented in Petit et al. (2015). A known mesh of the object is assumed and its pose and deformation are tracked with an RGB-D sensor. They demonstrated accurate results with a reasonable computational cost, at around 35 Hz.

Another task that requires real-time estimation of deformable objects is soft-tissue tracking, which is essential to automate surgical procedures such as minimally invasive surgeries. As minimally invasive surgeries, such as laparoscopic procedures, are preferred over open surgery as they have been shown to reduce recovery times for patients, there is an increasing need to improve the visualization of internal anatomy during surgery. One of the most successful approaches to address this issue has been the geometric reconstruction of soft tissue in three dimensions. For example, Malti et al. (2011) used shape-from-motion[15] to first construct a three-dimensional model (template) of a soft tissue before the surgeon caused any deformation. Once the surgeon started deforming the tissue, the three-dimensional shape was reconstructed based on monocular images as obtained by a laparoscopic camera. To account for the deformations caused by the surgeon, their approach considered shearing and anisotropy scaling while computing the correspondence between an image and the template. Their approach was able to produce decent results on in-vivo experiments. Similarly, dos Santos et al. (2014) proposed an algorithm to track soft-tissue deformations where, instead of relying on scaling parameters, they used surface matching to perform the registration. Their method proved to be accurate and fast enough to be used in clinical trials. As the body of works focused on three-dimensional registration for surgery assistance is beyond the scope of this survey, the interested reader is referred to Maier-Hein et al. (2013) for a thorough review of vision techniques dedicated to sensing soft tissue.

6.3. Deformation control

The methods presented in this section extend the technique first proposed by Wada et al. (1998), as reviewed in Section 4.2, to three-dimensional objects. These approaches have been applied to two different types of objects, namely *elastic objects* and *elasto-plastic objects*. Elastic objects return to their original shape once the external force is removed, and elasto-plastic objects are objects that present both plastic and elastic deformations (e.g. a loaf of bread). A commonality among these works is the use of a set of target points that are located inside the object which the robot cannot directly manipulate; thus, requiring an indirect manipulation of the points by deforming the object's surface.

6.3.1. Elasto-plastic objects. In Higashimori et al. (2010) a MSD model was used to predict the deformation of a clay-like object and modify its shape to a desired configuration. The approach consisted of two phases, namely, a *sensing* phase and a *shaping* phase. In the first phase, the object was pushed by an robotic actuator equipped with a force sensor to estimate the elasticity parameters of the object. Afterwards, the force required to reach the desired shape was computed using the obtained parameters and the integrated force applied during the sensing phase. The force was then applied in the shaping phase with a simple force feedback controller. In this approach, the deformation was only controlled in one dimension, i.e. it was deformed by a desired distance. Yoshimoto et al. (2011) extended this approach to deform the object in two dimensions by modeling the object as a connection of nodes in three-dimensional space.

6.3.2. Elastic objects. One of the first works concerning elastic, specifically hyperelastic, objects was reported in Smolen and Patriciu (2009). Based on a meshless model of the object they controlled a set of control points on the surface of the object through another set of points termed manipulation points. The control points were arbitrarily assigned to deform the object into a desired shape, whereas the manipulation points were located on the object's surface, i.e. where the manipulators could be positioned. Using these two set of points a Jacobian was computed to actuate the manipulators such that their motion drove the control points to a required configuration.

A series of reports by Navarro-Alarcón et al. used a similar approach, namely defining control and manipulation points and then commanding the motion of a robotic manipulator, through visual servoing, to reach a desired configuration for an elastic object. However, these works do not rely on a model of the object; instead, a vision system was used to track the control points. A distinctive characteristic of these works was the use of so-called *deformation feature* vectors based on a set of control points to describe the different types of deformation, including the following.

(a) *Point-based deformation:* The object is deformed to drive one point on the object to a desired target point.
(b) *Distance-based deformation:* For instance, moving the midpoint between two points a specified distance.
(c) *Angle-based deformation:* A line between two points on the object can be rotated as desired.
(d) *Curvature-based deformation:* An arc of three points on the object can be manipulated to achieve a specific curvature.

Figure 12 shows a graphic representation of these deformation features.

Furthermore, these approaches are based on estimating a Jacobian matrix representing the relation between the manipulator's motion and the feature vector. This Jacobian is termed deformation Jacobian, and is estimated online using the Broyden method[16] in Navarro-Alarcon et al. (2013a,b). In Navarro-Alarcon et al. (2014), they proposed a new online Jacobian estimator that uses views from multiple cameras. As these previous approaches only controlled three degrees of freedom of the manipulator's gripper, namely its position; in Navarro-Alarcon and Liu (2014), they extended their approach to incorporate the control of the gripper's orientation to have a six-degree-of-freedom controller. Another limitation of these approaches is that the control is performed on a two-dimensional projection based on the cameras' images, because the control points are described in pixel coordinates. In other words,

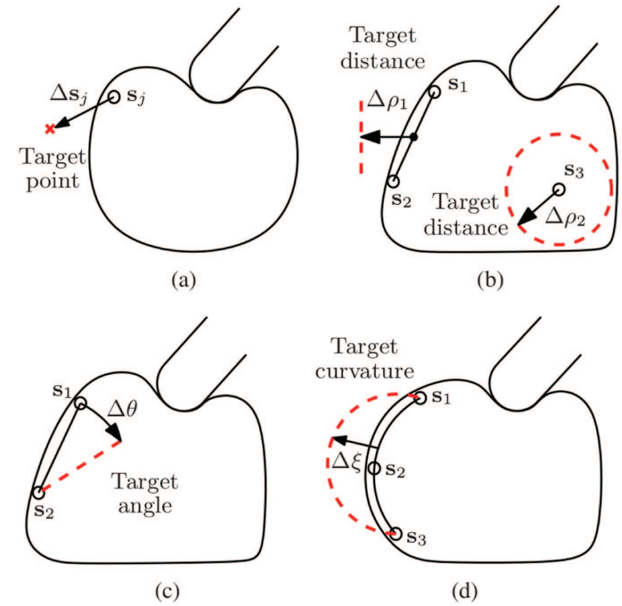

Fig. 12. Deformation features proposed by Navarro-Alarcon et al. (2013a).

this approach only controls the deformation features on a plane. This limitation was addressed in Navarro-Alarcon et al. (2016) by using stereo-vision to track the points in three dimensions and subsequently define the deformation feature vectors also in three dimensions.

The latest approach by Navarro-Alarcon and Liu (2017) proposed a different way of describing deformations by using a truncated Fourier series to approximate a contour shape, rather than a set of points. This novel representation is able to describe the contour of the object in a two-dimensional plane and it is used as a feedback signal to control the deformation of the object. As the method depends on the number of coefficients used to describe the object's two-dimensional shape, a trade-off between speed and accuracy can be adjusted depending on the application (e.g. a large number of harmonics can better describe the shape, but becomes computationally expensive to be used in real-time applications). One limitation of the approach, as it does not use a physical model, is its inability to guarantee if a desired shape can be reached in advance. Nevertheless, it would be interesting to research the viability of extending this innovative approach to three-dimensional shapes.

7. Discussion and future directions

7.1. Discussion

In the previous sections we reviewed recent approaches used in robot applications that manipulate deformable objects. These approaches were categorized according to the type of objects they manipulate, based on the categories described in Section 1, and subsequently depending on the manipulation task they perform upon the objects.

The top-level classification grouped the approaches into four categories of objects. Approaches used across these categories differ substantially, mainly due to the assumptions employed by the approaches. For instance, cloth-like objects can be represented as a polygonal structure, an assumption that has not been proven effective for solid objects. Another common assumption made by approaches used in the manipulation of DLOs is to describe them as a chain of links, which requires them to be extended to a grid in order to represent planar objects, as shown, for instance, in Elbrechter et al. (2011, 2012) and Namiki and Yokosawa (2015).

A further classification of the approaches was applied to distinguish them based on the manipulation task they focused on since the approach's success, and applicability, are highly dependent on the particular task at hand. Some tasks require high-accuracy estimation of the deformation (e.g. deformation control), whereas others do not have this constraint (e.g. picking up an object). Based on these differences, three main types of tasks were identified, namely:

(a) *Object grasping* is unarguably the most common manipulation task. Therefore, a vast amount of approaches have focused on grasping, picking up, and holding a object as a first attempt to manipulate deformable objects. This is a trend that has been consistent across all the object types, especially in the earliest approaches. In addition, grasping is usually a necessary first step to execute other manipulation actions such as lifting the object, folding a garment, or deforming the shape of an object. Another reason for the popularity of deformable object grasping might be related to the fact that rigid-body grasping is a very well-studied subject in robotics. So far, some approaches have tried to extend grasping techniques developed for rigid objects to deformable objects. For instance, the concept of form closure was extended to *deform closure* by Gopalakrishnan and Goldberg (2004); Gopalakrishnan (2005) to securely hold a deformable object; and force closure, in combination with an FEM simulation, was utilized by Lin et al. (2014) and Zaidi et al. (2017). Other approaches have developed new strategies, such as bulging the object enough for a manipulator to grasp it (Elbrechter et al., 2011; Kristek and Shell, 2012) or using machine learning to learn the force required to lift the object (Howard and Bekey, 1999).

(b) *Specific manipulation* tasks arise inherently due to the kind of object being manipulated. One specific task to DLO is to tie a knot, whereas for a cloth-like object, folding a garment is a particular task. Other specific tasks can be found in more than one category, as is the case of controlling the object deformation. Deformation control is a common task for both planar and solid objects.

(c) *Sensing* the state of a deformable object is a necessary requirement for most manipulation tasks. For instance, a manipulator deforming an object must be able to sense the object's shape to reach a desired shape. Sensing tasks can be classified in either *parameter identification* or *shape tracking*. Approaches focusing on parameter identification tasks use vision, and sometimes force data, to estimate either elasticity parameters (e.g. Young's modulus and Poisson's ratio) that are required by deformation models, such as, mass–spring models or FEM-based models; or approach-specific parameters, for instance, a degree of deformation required by an MSM simulation (Güler et al., 2015), the twisting strength and bending angles of a beam that are used in a Euler–Bernoulli model by Fugl et al. (2012), or the links' stiffness in a physics-based simulation of an object modeled as a series of rigid links (Caldwell et al., 2014). For shape tracking, there are approaches that rely only on vision systems to estimate the contour of the deformed object (Cretu et al., 2010a,b), while others combine deformation models with vision to estimate the object's shape (Leizea et al., 2014; Petit et al., 2015).

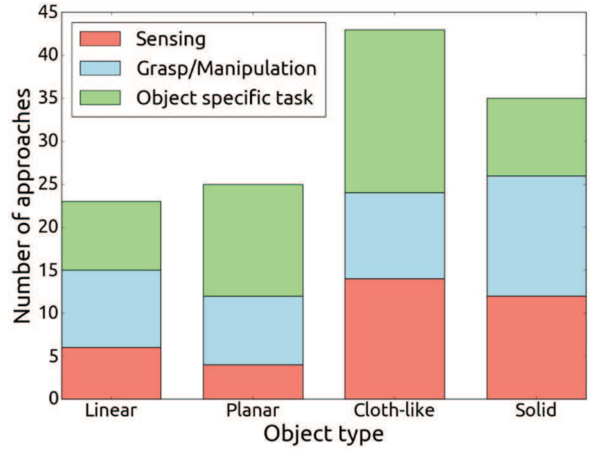

Fig. 13. Classification of the 106 reviewed approaches based on their task(s) and object type(s). Note that some approaches appear multiple times across task and object types.

Figure 13 shows the different tasks the reviewed approaches have focused on for each deformable object type. As shown in the figure, a larger number of approaches have been applied to cloth-like objects. This disparity might be due to the increment of recent projects focusing on solving laundry-related tasks with robots. Most notably, the UC Berkeley group[17] and the CloPeMa project[18] have produced a lot of results while performing many of the tasks required to enable a robot to autonomously do the laundry. The interest in solutions for laundry oriented tasks, within cloth-like objects, is also apparent in the number of approaches that solve a specific task (19) such as folding and hanging, in comparison with approaches focusing on manipulation (10) or sensing tasks (14).

Another notable feature from Figure 13 is the limited number of approaches focusing on sensing tasks for planar objects. In contrast to this is the number of approaches that focus on sensing tasks for solid objects. This is not a surprising fact, because our survey has mostly concentrated on reviewing recent approaches and, as noted previously, sensing tasks tend to precede manipulation tasks. Therefore, there are significantly fewer approaches focusing on sensing deformation on planar objects (which themselves have preceded approaches on solid objects) than on solid objects, which were not as frequently researched until the last few years.

7.2. Future directions

Finally, we discuss open problems within each category and suggest possible directions to be further researched.

7.2.1. Linear objects. A particular issue that has not been addressed substantially is that of estimating the state of linear deformable objects when three-dimensional deformations occur. This might be due to the fact that the majority of reviewed approaches either make assumptions on the object's behavior (e.g. assuming a rope's shape follows the trajectory of the gripper grasping it as in Vinh et al. (2012), Yamakawa et al. (2010), Takizawa et al. (2015), and Yamakawa et al. (2012)) or simplifying the object's configuration to, for instance, a topological description[19] as described by Yamakawa et al. (2008) and Matsuno et al. (2006). Nonetheless, there have been approaches attempting to estimate the shape of a linear deformable object. These approaches either use a modeling framework (Bretl and McCarthy, 2014), sensing (Matsuno et al., 2006), or, more commonly, a combination of both (Borum et al., 2014; Caldwell et al., 2014; Javdani et al., 2011).

However, the current state of the art on the state estimation of deformable objects is limited. On the one hand, modeling approaches require a priori knowledge of the object (e.g. elasticity parameters) and tend to be computationally expensive, rendering too slow for real-time manipulation tasks. On the other hand, sensing approaches thus far have mainly relied on visual sensors, a sensing modality that is affected greatly by occlusion, which occurs more often when manipulating the object. Other sensors, such as force–torque sensors, are currently limited by their sensitivity and might not be able to detect a significant measurement on objects that present large strains. Tactile sensing could be an interesting complementary modality to visual sensing due to its ability to extract relevant information on the parts of the objects that are being occluded by the manipulation process, although the resolution of a given tactile sensor restricts the sensing of sufficiently thin linear objects.

An accurate estimation of an object's state could allow more complex manipulation actions such as routing cables (e.g. in a car manufacturing facility) and helping disable people tie their shoelaces. In addition, improved sensing capabilities could detect unexpected events to trigger a replanning strategy. This sensing ability will benefit path planning approaches as it would allow them to adapt to dynamic environments; and understanding, either by modeling or sensing, how an object behaves while being manipulated, could be incorporated into planning strategies to generate more robust motions.

7.2.2. Planar objects. Most planar objects that have been studied so far are a simplified case of a solid object, where only one side of the object is considered. For instance, in Guo et al. (2013) a sponge-like object is manipulated while considering only its deformation on a two-dimensional plane. This was a necessary simplification at the beginning of the field, because the computational resources were not as powerful as they are today. By considering one dimension less of the object, simulation approaches could be applied that might not have been possible on a solid object as was demonstrated in Jia et al. (2011) and Gopalakrishnan (2005).

The state-of-the art computing is now, however, capable of simulating deformation of solid objects at an acceptable speed. Thus, the field has focused its attention on other types of objects such as paper. Paper manipulation remains an under-researched field, but there have been notable efforts to improve robots' abilities to handle paper (Elbrechter et al., 2011, 2012; Namiki and Yokosawa, 2015). Similar objects, e.g. cards and bills, have received even less attention and the current approaches focus on simple manipulation actions such as flipping them or rotating them (Kristek and Shell, 2012). Developing control algorithms that can be coupled with recent vision-based approaches, able to reconstruct deformations of planar surfaces in an online manner (see, for instance, Famouri et al., 2017), would provide robots with capabilities to manipulate paper-like objects in real time. This would allow robots, for example, to take over tedious office tasks such as sorting documents, making copies, opening mail, and handling bills.

Figure 13 indicates a clear trending on approaches focusing more on manipulating planar objects rather than on sensing their deformation. Most recent approaches have relied on estimating the object's deformation through modeling (either using mass–spring or FEM-based models) and apply control theory algorithms to manipulate them. A potential improvement to these techniques would be to add sensing information (e.g. force and vision) in the loop to increase the accuracy of both the deformation estimation and its control.

7.2.3. Cloth-like objects. Most of the focus so far has been on approaches to solve the so-called laundry problem; that is, picking up a cloth, place it into a washing machine/drier, taking it out of the washing machine/drier, stretching it out, folding it, and putting it away. However, more complicated tasks involving handling of clothes have received less attention. For instance, hanging a cap on a standing hanger was covered in Twardon and Ritter (2015) and an approach to iron a shirt with a robot was recently proposed by Li et al. (2016). Machine learning algorithms, already used for solving folding tasks, such as LfD (Balaguer and Carpin, 2011; Huang et al., 2015; Lee et al., 2015) could be applied to some of these complex tasks. Another example is that of aiding a person getting dressed; an initial attempt was presented by Yamazaki et al. (2014) and Tamei et al. (2011). However, helping a person button (and unbutton) a shirt proves to be a much more difficult task for a robot. Another desired capability for a robot is that of drying a disabled person using a towel, a task that does not only require the manipulation of the cloth itself, but also the interaction with the person's body which should be performed in a safe manner. Handling safety concerns opens the possibility of incorporating techniques currently being developed in the field of physical human–robot interaction.

7.2.4. Solid objects. This type of deformable objects remains the least researched. A reason for this is the computation complexity required to model a three-dimensional object in a realistic manner. However, due to recent computing progress that has allowed complex modeling and more realistic simulations to be performed at a reasonable rate; solid objects are now receiving more attention from the robotics community.

In so far, the research of solid deformable objects has focused mostly on grasping and holding an object (Lin et al., 2014, 2015; Zaidi et al., 2017), and on estimating its deformation (Leizea et al., 2014; Petit et al., 2015) (see Figure 13). Another direction that remains under-researched is that of *shape-servoing*, that is, to deform an object into a desired shape (Navarro-Alarcon et al., 2014, 2013b; Smolen and Patriciu, 2009). However, the current approaches that have focused on shape-servoing rely on visual sensing; which, similarly to the linear objects, are widely affected by occlusion and thus, adding force and tactile sensing could address the shortcomings of vision systems. Unlike vision sensing, force and tactile sensing require the robot to physically interact with the object which in turns require a manipulation strategy to perform different exploratory actions such as tapping, rubbing, shaking or grasping an object. Furthermore, these sensing modalities have demonstrated their ability to extract material properties to improve the object's deformation model.

An additional task with a lot of potential is that of dynamic manipulation of soft objects. For instance, a robot able to manipulate soft tissue in real time could assist a surgeon in a medical procedure, or help out in a domestic setting by handling food items to prepare a meal. As an example, one recent attempt of dynamic food manipulation was described by Satici et al. (2016), who proposed a theoretical framework to dynamically manipulate a pizza dough, specifically to toss and catch the dough with a humanoid robot. However, their results are yet to be validated on a real platform. Improvements on sensing, modeling, and actuation are still required to allow robots to handle deformable objects in a human-like manner. Accurate, and real-time, sensing would allow the robot to know the state of the object to better decide how to manipulate it to achieve a given task. Dexterous and fast manipulators are also necessary to carry on movements that adapt to an object constantly changing its shape.

Soft robots, unlike traditional robots, are made of materials exhibiting large strains (e.g. low Young's modulus) and could very well be considered deformable objects. The field of soft robotics shares several issues with deformable objects. For instance, soft robotics is concerned with estimating the robot's deformation and controlling its configuration by actions that deform the robot. In addition, methods such as FEM simulation are common, on both fields, to model the object/robot's behavior and, due to their intrinsic safety, soft robots are also ideal for manipulation tasks where safety is paramount in handling fragile objects (e.g. muscles and organs in a surgical operation or humans themselves while providing elderly care). As soft robotics covers a wide diversity of topics such as design, fabrication, modeling, and control, we refer the reader to previous surveys (e.g. Kim et al., 2013; Rus and Tolley, 2015; Trivedi et al., 2008) where these are discussed in depth.

8. Conclusion

In this survey, we have presented a review of recent approaches that focus on robot applications that manipulate deformable objects in domestic and industrial environments. A categorization of the approaches based on the type of objects they manipulate was defined, as well as a further classification that considers the type of task being executed. We have identified four object types, (1) *linear*, (2) *planar*, (3) *cloth-like*, and (4) *solid*, and three types of tasks, namely, sensing, manipulation, and tasks particular to each object type (e.g. folding a shirt). Using this classification, we have arranged the reviewed approaches in tables to offer a concise overview of the proposed categories.

We have also discussed current limitations based on this classification and provided prospective research lines for each type of object. We recognize that deformation sensing, mostly based on vision systems, has seen important advances recently. In comparison, manipulation skills still remain underdeveloped, in particular, dynamic manipulation of deformable objects. Control algorithms, and perhaps improved hardware, need to be developed to improve end-effectors' dexterity such that they can appropriately react to the object's deformation in real time. As manipulating the object might cause occlusions, vision-based sensing could overcome this by including the output of force and tactile sensors. This, in turn, requires deformation and contact models that transform force and tactile signals into a compatible representation (e.g. geometric information) obtained by visual systems.

Another insight, obtained by reviewing and classifying the state of the art in this field, is a clear tendency of approaches to develop techniques focused on specific tasks, or by exploiting assumptions that are unique to a given object type. Thus, a lack of approaches that provide general solutions for different object types and tasks remains a major open issue.

Finally, we intend for this survey to help researchers new to the field to obtain an overview of the state-of-the-art techniques and their applications, and provide researchers already in the field with a quick guide to relevant approaches on their particular interests.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has been sponsored by the French government research program Investissements d'Avenir through the RobotEx Equipment of Excellence (grant number ANR-10-EQPX-44) and the IMobS3 Laboratory of Excellence (grant number ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2007–2013 (ERDF - Auvergne region), and by the Auvergne region.

ORCID iD

Youcef Mezouar, https://orcid.org/0000-0001-8138-3928

Notes

1. For linear elasticity, large strain implies a low Young's modulus, e.g. less than 10 MPa (see Section 2 for a definition of Young's modulus).
2. We do not consider deformations produced by other physical phenomena such as temperature.
3. Linear elastic deformation occurs when the stress and strain are proportional (Callister, 2006).
4. We only consider physically-based deformation models.
5. An image registration algorithm that can be used for non-rigid registration (i.e. to track a deformable object) (Yang, 2011).
6. The planning algorithm is based on the covariant Hamiltonian optimization and motion planning (CHOMP) method (Ratliff et al., 2009).
7. A chart is an invertible map between a simple space (e.g. a three-dimensional Euclidean space) and a subset of a manifold (Morita, 2001).
8. The squeeze depth is the distance traveled by the fingers along a direction that produces either a stable or a pure squeeze.
9. Their simulated evaluation used 12–36 manipulation points.
10. A stack here refers to the stack created by a previous fold. For instance, a towel folded by the middle would result in a stack of two.
11. In this context, topological coordinates refer to relationships between segments as introduced in Ho and Komura (2009).
12. A slip was considered if a vision system detected that the object was not longer being grasped.
13. Volumetric mesh with known elasticity parameters such as Young's modulus and Poisson's ratio.
14. Unlike FEM and MSD models, position-based simulation are geometrically-based rather than physics-based. They trade off accuracy for speed in simulation.
15. Shape-from-motion is a technique that first generates a three-dimensional template of the target shape. Then, based on a set of points, it finds the correspondence between images, as the camera moves, and the previously generated template.
16. The Broyden method computes the Jacobian once at the beginning and then approximates it at each iteration using the previous Jacobian and the changes of the feature vectors and the end-effector's pose. In this way, the computed Jacobian relates the changes of the end-effector's pose with the deformation feature vector.
17. See https://people.eecs.berkeley.edu/~pabbeel/personal_robotics.html
18. See http://www.clopema.eu/
19. One example of a topological description is to represent the object as a graph, where each node represents either an end of a rope or an intersection, and links represent the rope segments between the nodes.

References

Acker J and Henrich D (2003) Manipulating deformable linear objects: characteristic features for vision-based detection of contact state transitions. In: *2003 IEEE International Symposium on Assembly and Task Planning*, pp. 204–209.

Alvarez N and Yamazaki K (2016) An interactive simulator for deformable linear objects manipulation planning. In: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 259–267.

Askeland D and Fulay P (2005) Mechanical Properties and Behavior. In: *The Science and Engineering of Materials* (6th edn). Cengage Learning.

Balaguer B and Carpin S (2011) Combining imitation and reinforcement learning to fold deformable planar objects. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 1405–1412.

Balkcom DJ and Mason MT (2008) Robotic origami folding. *The International Journal of Robotics Research* 27(5): 613–627.

Berenson D (2013) Manipulation of deformable objects without modeling and simulating deformation. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4525–4532.

Bergstrom N, Ek CH, Kragic D, Yamakawa Y, Senoo T and Ishikawa M (2012) On-line learning of temporal state models for flexible objects. In: *2012 IEEE-RAS International Conference on Humanoid Robots*, pp. 712–718.

Bersch C, Pitzer B and Kammel S (2011) Bimanual robotic cloth manipulation for laundry folding. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1413–1419.

Bicchi A (1995) On the closure properties of robotic grasping. *The International Journal of Robotics Research* 14(4): 319–334.

Boonvisut P and Cavusoglu MC (2013) Estimation of soft tissue mechanical parameters from robotic manipulation data. *IEEE/ASME Transactions on Mechatronics* 18(5): 1602–1611.

Boonvisut P and Cavusoglu MC (2014) Identification and active exploration of deformable object boundary constraints through robotic manipulation. *The International Journal of Robotics Research* 33(11): 1446–1461.

Borum A, Matthews D and Bretl T (2014) State estimation and tracking of deforming planar elastic rods. In: *2014 IEEE International Conference on Robotics and Automation*, pp. 4127–4132.

Bretl T and McCarthy Z (2014) Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research* 33(1): 48–68.

Buckingham R, Graham A, Arnarson H, Snaeland P and Davey P (2001) Robotics for de-heading fish - a case study. *Industrial Robot: An International Journal* 28(4): 302–309.

Caldwell TM, Coleman D and Correll N (2014) Optimal parameter identification for discrete mechanical systems with application to flexible object manipulation. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 898–905.

Callister WDJ (2006) Mechanical properties of metals. In: *Materials Science and Engineering: An Introduction* (7th edn). New York: Wiley.

Chen JS, Pan C, Wu CT and Liu WK (1996) Reproducing kernel particle methods for large deformation analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering* 139(1–4): 195–227.

Chen TL, Ciocarlie M, Cousins S, et al. (2013) Robots for humanity: using assistive robotics to empower people with disabilities. *IEEE Robotics and Automation Magazine* 20(1): 30–39.

Cretu AM, Payeur P and Petriu EM (2012) Soft object deformation monitoring and learning for model-based robotic hand manipulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(3): 740–753.

Cretu AM, Petriu EM, Payeur P and Khalil FF (2010a) Deformable object segmentation and contour tracking in image sequences using unsupervised networks. In: *2010 Canadian Conference on Computer and Robot Vision*, pp. 277–284.

Cretu AM, Petriu EM, Payeur P and Khalil FF (2010b) Estimation of deformable object properties from shape and force measurements for virtualized reality applications. In: *2010 IEEE International Symposium on Haptic Audio-Visual Environments and Games*, pp. 168–173.

Cusumano-Towner M, Singh A, Miller S, O'Brien JF and Abbeel P (2011) Bringing clothing into desired configurations with limited perception. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 3893–3900.

Das J and Sarkar N (2010) Planning and control of an internal point of a deformable object. In: *2010 IEEE International Conference on Robotics and Automation*, pp. 2877–2882.

Das J and Sarkar N (2011) Autonomous shape control of a deformable object by multiple manipulators. *Journal of Intelligent and Robotic Systems* 62(1): 3–27.

Das J and Sarkar N (2013) Passivity-based target manipulation inside a deformable object by a robotic system with noncollocated feedback. *Advanced Robotics* 27(11): 861–875.

Delgado Á, Jara CA, Mira D and Torres F (2015) A tactile-based grasping strategy for deformable objects' manipulation and deformability estimation. In: *International Conference on Informatics in Control, Automation and Robotics*, pp. 369–374.

Delgado Á, Jara CA, Torres F and Mateo CM (2016) Control of robot fingers with adaptable tactile servoing to manipulate deformable objects. In: *Robot 2015: Second Iberian Robotics Conference*, vol. 417. New York: Springer, pp. 81–92.

dos Santos T, Seitel A, Kilgus T, et al. (2014) Pose-independent surface matching for intra-operative soft-tissue marker-less registration. *Medical Image Analysis* 18(7): 939–952.

Doumanoglou A, Kargakos A, Kim TK and Malassiotis S (2014) Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In: *2014 IEEE International Conference on Robotics and Automation*, pp. 987–993.

Elbrechter C, Haschke R and Ritter H (2011) Bi-manual robotic paper manipulation based on real-time marker tracking and physical modelling. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1427–1432.

Elbrechter C, Haschke R and Ritter H (2012) Folding paper with anthropomorphic robot hands using real-time physics-based modeling. In: *2012 IEEE-RAS International Conference on Humanoid Robots*, pp. 210–215.

Erickson Z, Clegg A, Yu W, Turk G, Liu CK and Kemp CC (2017) What does the person feel? Learning to infer applied forces during robot-assisted dressing. In: *IEEE International Conference on Robotics and Automation*, pp. 6058–6065.

Famouri M, Bartoli A and Azimifar Z (2017) Fast shape-from-template using local features. *Machine Vision and Applications*. DOI: 10.1007/s00138-017-0876-9.

Fanson R and Patriciu A (2010) Model based deformable object manipulation using linear robust output regulation. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 496–501.

Frank B, Schmedding R, Stachniss C, Teschner M and Burgard W (2010) Learning the elasticity parameters of deformable objects with a manipulation robot. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1877–1883.

Frank B, Stachniss C, Schmedding R, Teschner M and Burgard W (2014) Learning object deformation models for robot motion planning. *Robotics and Autonomous Systems* 62(8): 1153–1174.

Fugl AR, Jordt A, Petersen HG, Willatzen M and Koch R (2012) Simultaneous estimation of material properties and pose for deformable objects from depth and color images. In: *Lecture Notes in Computer Science*, vol. 7476. Berlin: Springer, pp. 165–174.

Gemici MC and Saxena A (2014) Learning haptic representation for manipulating deformable food objects. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 638–645.

Gopalakrishnan K and Goldberg K (2004) D-space and deform closure: a framework for holding deformable parts. In: *2014 IEEE International Conference on Robotics and Automation* vol. 1, pp. 345–350.

Gopalakrishnan KG (2005) D-space and deform closure grasps of deformable parts. *The International Journal of Robotics Research* 24(11): 899–910.

Güler P, Pauwels K, Pieropan A, Kjellstrom H and Kragic D (2015) Estimating the deformability of elastic materials using optical flow and position-based dynamics. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots*, pp. 965–971.

Guo F, Lin H and Jia YB (2013) Squeeze grasping of deformable planar objects with segment contacts and stick/slip transitions. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 3736–3741.

Henrich D and Wörn H (2000) Robot manipulation of deformable objects. In: *Advanced Manufacturing* (1st edn). London: Springer.

Higashimori M, Yoshimoto K and Kaneko M (2010) Active shaping of an unknown rheological object based on deformation decomposition into elasticity and plasticity. In: *2010 IEEE International Conference on Robotics and Automation*, pp. 5120–5126.

Hirai S, Tsuboi T and Wada T (2001) Robust grasping manipulation of deformable objects. In: *2001 IEEE International Symposium on Assembly and Task Planning*, pp. 411–416.

Ho ES and Komura T (2009) Character motion synthesis by topology coordinates. *Computer Graphics Forum* 28(2): 299–308.

Howard A and Bekey G (1999) Intelligent learning for deformable object manipulation. In: *1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 15–20.

Hu AP, Bailey J, Matthews M, McMurray G and Daley W (2012) Intelligent automation of bird deboning. In: *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 286–291.

Huang SH, Pan J, Mulcaire G and Abbeel P (2015) Leveraging appearance priors in non-rigid registration, with application to manipulation of deformable objects. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 878–885.

Iyer S, Looi T and Drake J (2013) A single arm, single camera system for automated suturing. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 239–244.

Jackson RC, Desai V, Castillo JP and Cavusoglu MC (2016) Needle–tissue interaction force state estimation for robotic surgical suturing. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3659–3664.

Javdani S, Tandon S, Tang J, O'Brien JF and Abbeel P (2011) Modeling and perception of deformable one-dimensional objects. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 1607–1614.

Jia YB, Guo F and Lin H (2014) Grasping deformable planar objects: Squeeze, stick/slip analysis, and energy-based optimalities. *The International Journal of Robotics Research* 33(6): 866–897.

Jia YB, Guo F and Tian J (2011) On two-finger grasping of deformable planar objects. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 5261–5266.

Jia YB, Lin H and Guo F (2013) Optimal two-finger squeezing of deformable objects. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3514–3519.

Jimenez P (2012) Survey on model-based manipulation planning of deformable objects. *Robotics and Computer-Integrated Manufacturing* 28(2): 154–163.

Kampouris C, Mariolis I, Peleka G, et al. (2016) Multi-sensorial and explorative recognition of garments and their material properties in unconstrained environment. In: *2016 IEEE International Conference on Robotics and Automation*, June, pp. 1656–1663.

Khalil FF, Curtis P and Payeur P (2010) Visual monitoring of surface deformations on objects manipulated with a robotic hand. In: *2010 IEEE International Workshop on Robotic and Sensors Environments*, pp. 1–6.

Khalil FF and Payeur P (2010) Dexterous robotic manipulation of deformable objects with multi-sensory feedback - a review. In: Jimenez A and Hadithi BMA (eds.), *Robot Manipulators Trends and Development*. InTech, pp. 587–621.

Kim S, Laschi C and Trimmer B (2013) Soft robotics: A bioinspired evolution in robotics. *Trends in Biotechnology* 31(5): 287–294.

Kinio S and Patriciu A (2012) A comparative study of H∞ and PID control for indirect deformable object manipulation. In: *2012 IEEE International Conference on Robotics and Biomimetics*, pp. 414–420.

Kita Y, Kanehiro F, Ueshiba T and Kita N (2011) Clothes handling based on recognition by strategic observation. In: *2011 IEEE-RAS International Conference on Humanoid Robots*, pp. 53–58.

Kristek SM and Shell DA (2012) Orienting deformable polygonal parts without sensors. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 973–979.

Kruse D, Radke RJ and Wen JT (2015) Collaborative human–robot manipulation of highly deformable materials. In: *2015 IEEE International Conference on Robotics and Automation*, pp. 3782–3787.

Kudoh S, Gomi T, Katano R, Tomizawa T and Suehiro T (2015) In-air knotting of rope by a dual-arm multi-finger robot. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6202–6207.

Lamiraux F and Kavraki LE (2001) Planning paths for elastic objects under manipulation constraints. *The International Journal of Robotics Research* 20(3): 188–208.

Lee AX, Huang SH, Hadfield-Menell D, Tzeng E and Abbeel P (2014) Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4402–4407.

Lee AX, Lu H, Gupta A, Levine S and Abbeel P (2015) Learning force-based manipulation of deformable objects from multiple demonstrations. In: *2015 IEEE International Conference on Robotics and Automation*, pp. 177–184.

Leizea I, Alvarez H, Aguinaga I and Borro D (2014) Real-time deformation, registration and tracking of solids based on physical simulation. In: *2014 IEEE International Symposium on Mixed and Augmented Reality*, pp. 165–170.

Leonard S, Shademan A, Kim Y, Krieger A and Kim PCW (2014a) Smart Tissue Anastomosis Robot (STAR): Accuracy evaluation for supervisory suturing using near-infrared fluorescent markers. In: *2014 IEEE International Conference on Robotics and Automation*, pp. 1889–1894.

Leonard S, Wu KL, Yonjae Kim, Krieger A and Kim PCW (2014b) Smart Tissue Anastomosis Robot (STAR): a vision-guided robotics system for laparoscopic suturing. *IEEE Transactions on Biomedical Engineering* 61(4): 1305–1317.

Li Y, Hu X, Xu D, Yue Y, Grinspun E and Allen P (2016) Multi-sensor surface analysis for robotic ironing. In: *2016 IEEE International Conference on Robotics and Automation*, pp. 5670–5676.

Li Y, Wang Y, Case M, Chang SF and Allen PK (2014) Real-time pose estimation of deformable objects using a volumetric approach. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1046–1052.

Li Y, Xu D, Yue Y, Wang Y, Chang SF, Grinspun E and Allen PK (2015a) Regrasping and unfolding of garments using predictive thin shell modeling. In: *2015 IEEE International Conference on Robotics and Automation*, June, pp. 1382–1388.

Li Y, Yue Y, Xu D, Grinspun E and Allen PK (2015b) Folding deformable objects using predictive simulation and trajectory optimization. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6000–6006.

Lin H, Guo F, Wang F and Jia YB (2014) Picking up soft 3D objects with two fingers. In: *2014 IEEE International Conference on Robotics and Automation*, pp. 3656–3661.

Lin H, Guo F, Wang F and Jia YB (2015) Picking up a soft 3D object by "feeling" the grip. *The International Journal of Robotics Research* 34(11): 1361–1384.

Long P, Khalil W and Martinet P (2014a) Force/vision control for robotic cutting of soft materials. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4716–4721.

Long P, Khalil W and Martinet P (2014b) Robotic deformable object cutting: from simulation to experimental validation. In: *European Workshop on Deformable Object Manipulation*. Available at: http://www.irccyn.ec-nantes.fr/~martinet/publis/2014/ewdom14_long.pdf.

Lui WH and Saxena A (2013) Tangled: Learning to untangle ropes with RGB-D perception. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 837–844.

Maier-Hein L, Mountney P, Bartoli A, et al. (2013) Optical techniques for 3D surface reconstruction in computer-assisted laparoscopic surgery. *Medical Image Analysis* 17(8): 974–996.

Maitin-Shepard J, Cusumano-Towner M, Lei J and Abbeel P (2010) Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In: *2010 IEEE International Conference on Robotics and Automation*, pp. 2308–2315.

Malti A, Collins T and Bartoli A (2011) Template-based conformal shape-from-motion from registered laparoscopic images. *Medical Image Understanding and Analysis* 1(2): 1–5.

Matsuno T, Tamaki D, Arai F and Fukuda T (2006) Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information. *IEEE/ASME Transactions on Mechatronics* 11(4): 401–408.

McConachie D and Berenson D (2016) Bandit-based model selection for deformable object manipulation. In: *Workshop on the Algorithmic Foundations of Robotics*

Miller A and Allen P (2004) GraspIt! *IEEE Robotics and Automation Magazine* 11(4): 110–122.

Miller S, Van Den Berg J, Fritz M, Darrell T, Goldberg K and Abbeel P (2012) A geometric approach to robotic laundry folding. *The International Journal of Robotics Research* 31(2): 249–267.

Moll M and Kavraki L (2004) Path planning for minimal energy curves of constant length. In: *2004 IEEE International Conference on Robotics and Automation*, pp. 2826–2831.

Moll M and Kavraki L (2006) Path planning for deformable linear objects. *IEEE Transactions on Robotics* 22(4): 625–636.

Monsó P, Alenyà G and Torras C (2012) POMDP approach to robotized clothes separation. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1324–1329.

Moore P and Molloy D (2007) A survey of computer-based deformable models. In: *International Machine Vision and Image Processing Conference (IMVIP 2007)*, pp. 55–66.

Morita S (2001) Manifolds. *Geometry of Differential Forms* 201: 1–55.

Namiki A and Yokosawa S (2015) Robotic origami folding with dynamic motion primitives. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 5623–5628.

Navarro-Alarcon D and Liu Yh (2014) A dynamic and uncalibrated method to visually servo-control elastic deformations by fully-constrained robotic grippers. In: *2014 IEEE International Conference on Robotics and Automation*, pp. 4457–4462.

Navarro-Alarcon D and Liu YH (2017) Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-D image contours. *IEEE Transactions on Robotics*. DOI: 10.1109/TRO.2017.2765333.

Navarro-Alarcon D, Liu YH, Romero JG and Li P (2013a) Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Transactions on Robotics* 29(6): 1457–1468.

Navarro-Alarcon D, Liu Yh, Romero JG and Li P (2014) On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments. *The International Journal of Robotics Research* 33(11): 1462–1480.

Navarro-Alarcon D, Yip HM, Wang Z, et al. (2016) Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model. *IEEE Transactions on Robotics* 32(2): 429–441.

Navarro-Alarcon D, Yunhui Liu, Romero JG and Peng Li (2013b) Visually servoed deformation control by robot manipulators. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 5259–5264.

Nealen A, Müller M, Keiser R, Boxerman E and Carlson M (2006) Physically based deformable models in computer graphics. *Computer Graphics Forum* 25(4): 809–836.

Nguyen VD (1988) Constructing force-closure grasps. *The International Journal of Robotics Research* 7(3): 3–16.

Petit A, Lippiello V and Siciliano B (2015) Real-time tracking of 3D elastic objects with an RGB-D sensor. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3914–3921.

Rambow M, Schauss T, Buss M and Hirche S (2012) Autonomous manipulation of deformable objects based on teleoperated demonstrations. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2809–2814.

Ramirez-Alpizar IG, Higashimori M, Kaneko M, Tsai CHD and Kao I (2012) Dynamic nonprehensile manipulation for rotating a thin deformable object: An analogy to bipedal gaits. *IEEE Transactions on Robotics* 28(3): 607–618.

Ramisa A, Alenya G, Moreno-Noguer F and Torras C (2012) Using depth and appearance features for informed robot grasping of highly wrinkled clothes. In: *2012 IEEE International Conference on Robotics and Automation*, pp. 1703–1708.

Ratliff N, Zucker M, Bagnell JA and Srinivasa S (2009) CHOMP: Gradient optimization techniques for efficient motion planning. In: *2009 IEEE International Conference on Robotics and Automation*, pp. 489–494.

Roussel O and Ta M (2014) Deformable linear object manipulation planning with contacts. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–5.

Rus D and Tolley MT (2015) Design, fabrication and control of soft robots. *Nature* 521(7553): 467–475.

Saadat M and Nan P (2002) Industrial applications of automatic manipulation of flexible materials. *Industrial Robot: An International Journal* 29(5): 434–442.

Satici AC, Ruggiero F, Lippiello V and Siciliano B (2016) A coordinate-free framework for robotic pizza tossing and catching. In: *2016 IEEE International Conference on Robotics and Automation*, pp. 3932–3939.

Schulman J, Lee A, Ho J and Abbeel P (2013) Tracking deformable objects with point clouds. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 1130–1137.

Shah AJ and Shah JA (2016) Towards manipulation planning for multiple interlinked deformable linear objects. In: *2016 IEEE International Conference on Robotics and Automation*, pp. 3908–3915.

Shibata M, Ota T and Hirai S (2009) Wiping motion for deformable object handling. In: *2009 IEEE International Conference on Robotics and Automation*, pp. 134–139.

Sin FS, Schroeder D and Barbič J (2013) Vega: non-linear FEM deformable object simulator. *Computer Graphics Forum* 32(1): 36–48.

Smolen J and Patriciu A (2009) Deformation planning for robotic soft tissue manipulation. In: *International Conferences on Advances in Computer–Human Interactions*, pp. 199–204.

Stria J, Prusa D, Hlavac V, et al. (2014) Garment perception and its folding using a dual-arm robot. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 61–67.

Sun L, Aragon-Camarasa G, Rogers S and Siebert JP (2015) Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In: *2015 IEEE International Conference on Robotics and Automation*, pp. 185–192.

Takizawa M, Kudoh S and Suehiro T (2015) Method for placing a rope in a target shape and its application to a clove hitch. In: *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 646–651.

Tamei T, Matsubara T, Rai A and Shibata T (2011) Reinforcement learning of clothing assistance with a dual-arm robot. *2011 IEEE-RAS International Conference on Humanoid Robots* : 733–738.

Taylor RH, Menciassi A, Fichtinger G, Fiorini P and Dario P (2016) Medical robotics and computer-integrated surgery. In: *Springer Handbook of Robotics*. Cham: Springer, pp. 1657–1684.

Tian J and Jia YB (2010) Modeling deformations of general parametric shells grasped by a robot hand. *IEEE Transactions on Robotics* 26(5): 837–852.

Timoshenko S and Woinowsky-Krieger S (1959) *Theory of plates and shells* (2nd edn). New York: McGraw-Hill.

Tokumoto S, Fujita Y and Hirai S (1999) Deformation modeling of viscoelastic objects for their shape control. In: *1999 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 767–772.

Tonnesen DL (1998) *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction, and Animation.* PhD Thesis, University of Toronto.

Trivedi D, Rahn CD, Kier WM and Walker ID (2008) Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics* 5(3): 99–117.

Twardon L and Ritter H (2015) Interaction skills for a coat-check robot: identifying and handling the boundary components of clothes. In: *2015 IEEE International Conference on Robotics and Automation*, pp. 3682–3688.

Vinh TV, Tomizawa T, Kudoh S and Suehiro T (2012) A new strategy for making a knot with a general-purpose arm. In: *2012 IEEE International Conference on Robotics and Automation*, pp. 2217–2222.

Wada T, Hirai S and Kawamura S (1998) Indirect simultaneous positioning operations of extensionally deformable objects. In: *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1333–1338.

Weifu Wang, Berenson D and Balkcom D (2015) An online method for tight-tolerance insertion tasks for string and rope. In: *2015 IEEE International Conference on Robotics and Automation*, pp. 2488–2495.

Yamakawa Y, Namiki A and Ishikawa M (2010) Motion planning for dynamic knotting of a flexible rope with a high-speed robot arm. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 49–54.

Yamakawa Y, Namiki A and Ishikawa M (2011) Motion planning for dynamic folding of a cloth with two high-speed robot hands and two high-speed sliders. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 5486–5491.

Yamakawa Y, Namiki A and Ishikawa M (2012) 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*, pp. 2249–2254.

Yamakawa Y, Namiki A, Ishikawa M and Shimojo M (2008) Knotting manipulation of a flexible rope by a multifingered hand system based on skill synthesis. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2691–2696.

Yamazaki K (2014) Grasping point selection on an item of crumpled clothing based on relational shape description. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3123–3128.

Yamazaki K, Oya R, Nagahama K, Okada K and Inaba M (2014) Bottom dressing by a life-sized humanoid robot provided failure detection and recovery functions. *2014 IEEE/SICE International Symposium on System Integration*, pp. 564–570.

Yang J (2011) The thin plate spline robust point matching (TPS-RPM) algorithm: A revisit. *Pattern Recognition Letters* 32(7): 910–918.

Yoshida E, Ayusawa K, Ramirez-Alpizar IG, Harada K, Duriez C and Kheddar A (2015) Simulation-based optimal motion planning for deformable object. In: *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts*, pp. 1–6.

Yoshimoto K, Higashimori M, Tadakuma K and Kaneko M (2011) Active outline shaping of a rheological object based on plastic deformation distribution. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 1386–1391.

Yu W, Kapusta A, Tan J, Kemp CC, Turk G and Liu CK (2017) Haptic simulation for robot-assisted dressing. In: *IEEE International Conference on Robotics and Automation*.

Zaidi L, Corrales JA, Bouzgarrou BC, Mezouar Y and Sabourin L (2017) Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems* 95: 196–206.

Zhou D, Claffee MR, Lee KM and McMurray GV (2006a) Cutting, ‘by pressing and slicing’, applied to robotic cutting biomaterials, Part I: Modeling of stress distribution. In: *2006 IEEE International Conference on Robotics and Automation*, pp. 2896–2901.

Zhou D, Claffee MR, Lee KM and McMurray GV (2006b) Cutting, ‘by pressing and slicing’, applied to the robotic cut of biomaterials. II. Force during slicing and pressing cuts. In: *2006 IEEE International Conference on Robotics and Automation*, pp. 2256–2261.