Survey on Model-based Manipulation Planning of Deformable Objects

P. Jiménez

Institut de Robòtica i Informàtica Industrial (CSIC - UPC)
Llorens i Artigas 4-6, E-08028 Barcelona, Spain
e-mail: pjimenez@iri.upc.edu
Tel.: +3434015784; fax:+3434015750

Abstract

A systematic overview on the subject of model-based manipulation planning of deformable objects is presented. Existing modelling techniques of volumetric, planar and linear deformable objects are described, emphasizing the different types of deformation. Planning strategies are categorized according to the type of manipulation goal: path planning, folding/unfolding, topology modifications and assembly. Most current contributions fit naturally into these categories, and thus the presented algorithms constitute an adequate basis for future developments.

Key words: robotic manipulation, manipulation planning, deformable objects, deformable models

1. Introduction

Manipulation planning differs from standard motion planning in that the focus is not on the robot and its displacements but rather on the object(s) to be manipulated. Manipulation of *rigid* objects consists basically in changing their *pose* (position and orientation), while avoiding collisions, in the context of pick-and-place or assembly tasks. Realistic instances of the problem take also constraints on the stability of the grasp and of the placement of the object into account (see [1] for a small survey on the subject). When *deformable* objects are involved, manipulation will in most cases also affect their *shape*, with geometrical or topological changes. This behavior has to be considered in the whole planning process.

Most existing robotic systems that handle deformable objects rely heavily on a sensor-based control scheme. Many of them are ad-hoc algorithms, which lack generality [2]. Moreover, flexibility of such systems is also poor, as they implement fixed strategies. Sensorial feedback may be necessary in real implementations to overcome uncertainties in the model, which will probably arise except in very controlled environments. Furthermore, sensorial information is also needed in the automatic acquisition of behavior models of deformable objects under manipulation (i.e., machine learning of manipulation), a little explored issue. A collection of control-related articles can be found in [3].

Planning chooses and/or instantiates a sequence of actions from a given repertoire. This choice is not only conditioned by the intended sequence of states (in this case, configurations of the manipulated object) but also by the constraints that each state poses on the possible actions. Here, internal and external constraints have to be considered, as well as feasibility (hard) and optimization (soft) constraints. Constraints allow to narrow the search space of actions to consider, and to introduce criteria for evaluating different choices.

The present work concentrates on model-based off-line manipulation planning, which is based on geometrical (and to some extent also physical) descriptions of the object to be manipulated and its environment. We also assume that the wealth of possible sequences of elementary motions or actions makes it necessary to devise the manipulation plans automatically. Furthermore, some kind of prehensile grasping is involved in manipulation (see [4] for nonprehensile manipulation). In the works referenced in our survey, simplified models of grasping are adopted, in the form of *manipulation constraints* that define the location of grasps on the object and tangents of its surface at these points, and assuming that the forces exerted by the fingers ensure the intended deformations without damaging the object.

Deformations depend on the material properties of the objects, on their initial shape and dimensions, and also on the localization, direction, intensity, duration and frequency of the applied forces. As for manipulation planning, there are two features that characterize object deformations: reversibility and direction/extension of the shape change. The first criterion refers to whether the object recovers completely its original (rest) shape once the external forces have ceased (**elastic** deformation), or if permanent, stable deformations appear on the body of the object (**plastic** deformation). A third category of deformations may be considered, as for reversibility, which is

2

applicable to material like cloth or rope. We call them **flexible** deformations, as they are neither elastic (due to friction and gravity, the object does not recover anything like a "rest shape" once manipulation ends) nor strictly plastic, as they can be altered with very slight effort. Such definition is not a standard one in materials science or deformation theory, but it is intuitive and useful for robotic planning purposes.

Deformations are also characterized by the direction in which they take place, as a result of the direction of the applied forces and moments. Figure 1 displays a schematic view of some basic deformation types for a cylinder.
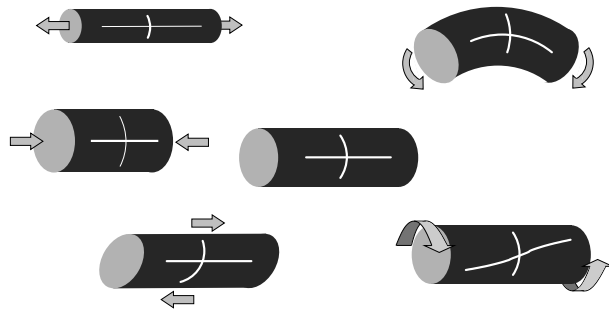

Figure 1: The undeformed cylinder appears in the center of the figure. Arrows stand for the forces and moments exerted on the cylinder, while the cross drawn on it helps to visualize the resulting deformation. At the left, tension, compression and shear, and on the right bending and torsion are exemplified.

2. Modelling deformable objects

Models aim at capturing the behaviour of the represented objects when they are manipulated. The scope and granularity of such models is obviously application-dependent, and a compromise between accuracy and computational efficiency is always sought. Unlike for rigid objects, in most cases the manipulation of deformable objects is quite tolerant to collisions (the shape adapts to the colliding object). This allows to do also simplifying assumptions about their shape.

2.1. General deformable volumetric models

The most distinguishing feature of deformable objects refers to the changes in shape that they experience under the influence of external and internal forces. From the point of view of planning, these shape modifications can

be the goal of planning itself (e.g., if they allow to avoid obstacles along a path) or a side-effect of a given planned action. In any case, there is an underlying physical behavior that governs these shape alterations in real three-dimensional objects. This physical behaviour can be implicitly considered while restricting to use pure geometric models (like **splines and patches** manipulated by their control points) or it can be explicitly reflected in the model. In the latter case, **physical analogues** like mass-spring models, particle systems or linked volumes, or more accurate **continuum models** may be used. Accurate models, like those associated with the Finite Element Method (FEM), are in fact more appropriate for off-line simulations. More recent approximate models meet both the requirements of realistic behavior and real time execution. Also **hybrid models** exist that combine some of the features of these different types. Excellent surveys exist on this subject in the field of Computer Graphics [5] as well as in Surgery Simulation [6], where not only elastic deformations but also incisions and suturing have to be considered. Considering the suitability for robotic purposes, the choice among the different types of models is a matter of available programming time and computational resources, as well as of the required accuracy. Physical analogues like mass-spring models are popular as they are relatively easy to implement and to tune, and provide a good compromise between accuracy (which can also be tuned, by varying the number of nodes and springs) and speed. However, tight tolerances may require the use of FEM, in an off-line pre-programming stage.

2.2. Deformable planar models

Planar objects have two privileged dimensions, whereas thickness, the third one, is negligible for manipulation planning purposes, besides grasping or collision detection. A paradigmatic example is sheet metal bending: thickness —together with material properties— plays an important role when determining the required punch displacement for a given bend angle in a bending machine, but it can be neglected when computing the necessary manipulation of the part during the bending process [7]. Roughly, one may distinguish between models that aim at reproducing continuous deformations, and models that operate on a more abstract level accounting for plastic deformations at prespecified discrete points. Among the first ones, a further distinction can be made depending on whether the material has a regular isotropic behavior or not.

Continuous deformations of uniform and isotropic material. Thin metal and plastic plates subject to elastic deformations can be easily modelled as a low-degree Bézier surface [8]. Its configuration is determined by the initial (load-free) shape, the deformation, and the rigid transformation. An energy model of the surface penalizes deformations that lead towards high curvatures, extension or shear of the surface, based exclusively on geometric parameters. In [9, 10] a more realistic elastic deformation model, which takes the material properties directly into account (Young modulus and Poisson ratio), is used. Together with the *grasping conditions* (a.k.a. *manipulation constraints*, see Section 5), that determine the location of specific control points and the corresponding tangents to the surface, elastic energy minimization determines the position of the other control points, and consequently the deformation of the plate. Similar models are used in [11]. The issue of extending such models to deal with permanent deformations is difficult and remains open.

Deformations of anisotropic material. Materials like cloth or fabric exhibit a complex behavior and are thus hard to model. Difficulties include the highly deformable nature of cloth, where subtle mechanical variations are amplified into large draping or motion variations, and its highly intricate anisotropic and nonlinear mechanical behavior [12]. Standard protocols, as the Kawabata Evaluation System (KES) [13], based on experimental measurement of strain-stress curves (elongation, bending and shearing), as well as surface properties, provide a basis for an adequate parameterization of the mechanical properties of different kinds of cloth (the fabric drape coefficient is related to the mechanical properties from KES tests in [14]). The Computer Graphics community has devoted large efforts to cloth modelling and simulation, since a couple of decades ago. Fabric has been modelled mainly with continuum models, implemented with finite difference [15] and finite elements methods [16–19], and with discrete models like particle systems [20, 21] and mass-spring models [12, 22–24]. In the latter, immediate neighbors are connected with "structural springs", diagonal neighbors with "shear springs" and also cross-springs connecting non-immediate neighbors are necessary for modelling the flexural resistance of cloth. Continuum models face serious drawbacks when applied to cloth, due to their high computational requirements (very fine meshing to produce large deformations) and difficult integration of highly variable constraints [12], mainly the high instability and non-linearity of the buckling behavior (formation of wrinkles) [22]. Discrete systems perform better as for the quality/efficiency tradeoff. As for numerical

integration methods, explicit methods are fast and easy to implement, but require small simulation timesteps for ensuring stability, whereas implicit methods circumvent this problem [12]. Viscosity (other than the *material intrinsic damping* of cloth [22]) is often added to the model to speed up convergence and to enhance numerical stability [12, 22]. Most methods relate cloth deformation to an energy function, although attempts at incorporating measurable mechanical properties directly into the model also exist [25]. All these findings are useful for the robotics community, and recent works make use of discrete models for image registration of non-rigid planar objects (fixed-length edge meshes for inextensible material [26] and weighted basis shape meshes that stand for admissible deformation modes for strechable surfaces [27]), as well as of continuum models for action selection [28].

Plastic deformations at discrete points. A fold in a sheet of paper, a crease in a carton, or a bend in a sheet metal blank is a plastic deformation and thus influences its shape and behavior permanently. Folding is a continuous motion that preserves the distances on the surfaces (no tear or stretch is allowed) and avoids self-intersections, although the surfaces may bend freely [29], and, in particular, may touch themselves (for example, flat foldings in flat origami). Objects experimenting such deformations can be treated as multilink bodies, i.e., rigid parts connected at rotational joints, often oriented at three orthogonal directions. The number of folds (joints) is typically low. Rather than computing a continuous path for such a structure between an initial (the flat sheet) and a goal configuration, planning refers to sequencing, i.e., to determine an ordering of the single bending operations. Each fold is bent only once, to its final fixed angular value. Planning does not occur in the configuration space of the joint values, but in the space of possible states of bending. Despite the high-level nature of such sequencing, it is directly conditioned by geometric constraints, like those imposed by collision avoidance. Thus, while the different states can typically be represented as nodes on a graph, labelled by the bends attained so far, the links between them will not only be determined by combinatorics but also by geometry: two states won't be connected if the bending that leads from one state to the other means a collision. Furthermore, the links in this state space representation can be weighted according to some cost criterion (Section 6).

2.3. Deformable linear models

Linear means that one of the dimensions of the object is clearly prevalent over the other two: cables, wires, threads, beams, hoses, ropes, tubes,

catheters and needles, among others, fall inside this category. They are referred to generically as *Deformable Linear Objects* (DLO).

Continuous deformations of elastic curves. Some DLOs are identified with the trajectory they follow in a given setting. This is the case of steerable flexible bevel-tip needles inserted in living tissue, which follow paths of piecewise constant curvature [30, 31]. A more explicit formulation relates geometrical features of the curve such as curvature and torsion to its energy. Minimum energy configurations of the curve under a number of manipulation constraints and assuming there is no stretch are computed [32, 33]. Alternatively, stress and strain of the curve can be directly linked, as in the most-used mass-spring models [10, 34–36], or the quite promising Cosserat models, that do account also for twisting behavior [37]. Linear spring–based lattices [10, 34] are still three-dimensional, whereas in pure linear models additional torsional and angular springs [35] are used or generalized spring models which react against bending and twisting force as well as stretching [36].

Flexible deformations. DLOs that present a highly flexible behavior, like threads, sutures, or ropes, can be easily modelled as linear arrangements of mass-nodes and springs [38, 39], chains of many straight rigid links connected by spherical joints [40], or chains of touching or partially overlapping spheres (see, for example, [39]).

Knot tying is presented as a case study of flexible object manipulation in [41, 42]. Here rather topology than the exact geometry of the object is a matter of concern. The state of the rope is represented as a sequence of *crossings*, as they would appear on a projection on the plane, from the left to the right endpoint of the rope (Figure 2).

Plastic deformations. Models do operate at a geometric level, where hard precedence constraints arising from collision detection are determined as in a kinematic chain, and a combinatorial level, where the precedence constraints between bending operations translate into the existence or not of links between the nodes that correspond to bending states, if a graph representation is chosen.

3. Collision detection

Collision detection strategies based on hierarchies of bounding volumes belong to the most popular ones, for their proven efficiency [44]. In the case of deformable objects, it makes no sense to care about tight fitting bounding
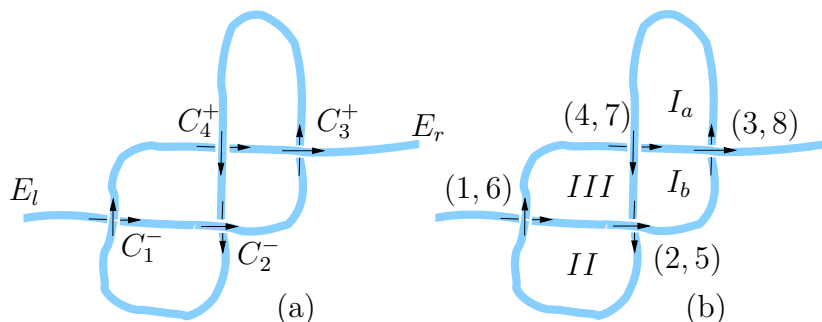
Figure 2: (a) Sequence of crossings representing the state of the rope: $E_l - C_1^{l-} - C_2^{u-} - C_3^{l+} - C_4^{u+} - C_2^{l-} - C_1^{u-} - C_4^{l+} - C_3^{u+} - E_r$. The first two crossings are *left-handed helical* (labelled with a minus sign) and the other two are *right-handed helical* (plus label). The labels u and l refer to the upper and lower part of the crossing, respectively. (b) An alternative representation, used in [43]. The numbering of the loops is conditioned by the forming sequence, which in this case is $(2, 5), (1, 6), (4, 7), (3, 8)$ (this corresponds to reversing the upper sequence shown later in Figure 9), note that the last step pierces loop I in two loops I_a and I_b.

volumes, which are computationally more costly, but simpler volumes whose overlapping is cheaper to test for are clearly preferred, like Axis-Aligned Bounding Boxes (AABB) [45–49]. The key point lies in deriving an efficient updating procedure. In [50], two other families of methods are discussed, namely stochastic methods and methods based on distance fields. As for planar objects, surface-volume collisions can be efficiently detected using again hierarchies of bounding volumes, like AABB hierarchies [51, 52], hierarchies of k-dops [23, 53, 54], or dynamic bounding volume hierarchies [55]. The more problematic surface-surface collisions (including self-collisions) have been addressed in [23, 51, 54, 56, 57]. Other type of collisions of planar objects to be analyzed are those arising from bending operations: the volumes swept out by the rotating parts around the fold are tested for intersection with other parts of the object (self-collisions) or with other objects in the environment. In the case of linear objects, an efficient hierarchical structure of spheres is maintained for performing efficient collision and self-collision detection [58].

4. Planning the manipulation of deformable objects

The type of manipulation goal constitutes a quite discriminating classification criterion as it groups together applications with similar constraints

and where manipulation planning is performed with similar algorithms. The following categories can be distinguished:

- **Path planning** Typically an initial and a goal configuration (including the shape of the object or of a specific part) are defined, and a collision-free path has to be found connecting them.

- **Folding/unfolding** Instances of the most elemental deformation operation (the fold) are combined to yield a specific shape specification. It deserves an own category due to the discrete and localized nature of the folds.

- **Modifying the topology** goes one step further, as for sophistication of the deformations. These changes are not geometrically precise, but qualitatively meaningful.

- **Assembly** in its most general meaning refers to the contact relationships established with objects in the environment. Constraints expressed as contacts between specific primitives of the involved objects have to be considered in planning.

5. Path planning for elastic objects

5.1. General formulation.

Standard path planning algorithms may be adapted to deal with elastic deformations. An important number of contributions opt for multiple-query planners like Probabilistic Roadmap Methods [8–11, 59–61]. The most direct way is to extend raw PRM strategy to the get random samples of configurations determined by the initial (load-free) shape, the deformation, and the rigid transformation, without explicitly considering manipulation constraints [8]. Samples whose energy values exceed predefined limits are discarded in the same way that colliding configurations. Aiming at a more realistic setting, the next step is to explicitly capture the grasping or manipulation constraints [9, 10, 32, 33, 59]. Here, first a random manipulation constraint is generated and tested for feasibility (both the plain strain limit and the curvature limit are checked for locally). Together with the grasping conditions, elastic energy minimization determines the position of the other control points, and consequently the deformation of the object. Then, for valid deformations, a given number of randomly generated rigid transformations are checked for

collisions with the obstacles. Further constraints on the shape can be given by fixing the position and tangent of an arbitrary number of intermediate control points, as done in [32, 33] for DLOs of constant length. The shape of each fragment of the curve is computed from the relationship between the internal energy and the torsion and curvature of the curve. As for the local planner, whose mission is to connect the new configurations to the existing roadmap, it works by computing first the necessary rigid transformations and then interpolating between the manipulation constraints to obtain a *deformation path*. A further enhancement is introduced in [11, 34], by using the medial axis of the workspace for guiding the sampling of the probabilistic roadmap method. Configurations of the object on or near the medial axis are obtained by fitting the manipulation constraints on it.

The approach followed in [60] differs from the previous ones in that the collision test of traditional PRM is replaced with an *acceptable penetration* (in C-space) test, the needed deformations to obtain a collision-free state are estimated and constitute the weight of the different roadmap links. A similar approach is taken in [61], where the roadmap is first computed while treating the deformable object as a point robot, a shortest path is found, and hard-constraints as non-penetration (in the obstacles) and volume preservation (using the ideal gas law) of the deformable object, together with soft constraints as path-following and energy minimization, guide the necessary deformations of the object. Single-query planners like Rapidly-exploring Random Tree (RRT) planners [62], Expansive Space Trees (EST) [63] or the very promising Sampling-based roadmap of trees (SRT) [64] could be adapted as well to deal with manipulation planning of deformable objects.

5.2. Needle Steering.

Steering elastic needles with bevel tips in soft tissue while avoiding obstacles —as needed in applications like suturing or endoscopic manipulation— can be done by controlling two degrees of freedom at the needle base (bevel direction and insertion depth). High torsional rigidity is assumed for this kind of needles, which means that an axial rotation on the needle shaft at the insertion point translates into an equal axial rotation of the tip. The asymmetry of the tip causes the needle to bend due to the forces exerted on the tip by the pierced tissue.

As shown in [31], assuming a stiff tissue, this kind of needle cuts a path of constant curvature in the direction of the bevel, and the needle shaft bends so as to follow this path. Thus, steering can be treated as a nonholonomic

motion planning problem for a Dubins car with no reversal [30]. In [65] the control variables include also the location of the insertion point and insertion angle. Simulations show that, whereas a frictionless needle inserted in a stiff tissue follows a path with constant curvature, this does no longer hold in the case of deformable tissue and considering the effect of friction forces. More research has to be devoted to this point, as well as to the presence of inhomogeneities in the tissue and to the behavior of different bevel tip designs. Full 3D simulation is also an open issue. A first step has been done in [66] the planning of the needle path extends to the 3D space (not restricted to the imaging plane) of an isotropic tissue without obstacles (Figure 3). Thus, both linear and angular insertion velocities are variables in the steering model.
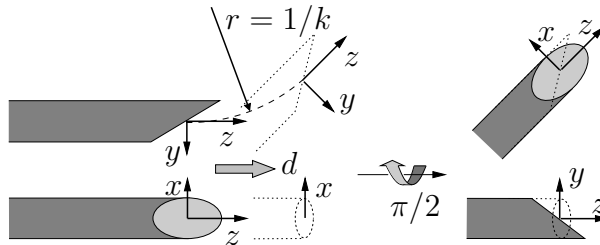

Figure 3: Due to torsional rigidity, the angular velocity at the insertion point causes the tip to rotate and thus to change the direction of progress. A 3D linear trajectory results for the needle. Two views are shown: a side view on top, and a view from below, before and after a given linear displacement and an axial rotation of $\pi/2$.

6. Folding/unfolding

The manipulated object is a workpiece which has to be deformed to a desired shape combining elemental folds, which have to be located at specific points. Two problems may be considered: where to fold (for example, in *origami design* [67–69]), and in which order. Although manipulability issues may be considered in the design phase, the functional requirements of the piece are certainly determinant as for the final location of the folds. But manipulability becomes really important when deciding in which order to fold. Sequencing of folding operations is a combinatorial problem, subject to hard constraints and optimality criteria. The feasibility of partial orderings of folds in this case is basically determined by collision constraints, including object-robot, object-environment, and robot-environment collisions as well

as selfcollisions. In the case of sheet metal bending, other hard constraints arise from tolerance specifications (see Section 6.4). As for soft optimality constraints, they are mostly concerned with minimizing the manipulation time and efforts. These hard and soft constraints may appear explicitly in the form of rules arising from expertise, or may be implicitly encoded in precedence constraints between pairs of bending operations. Graph representations and the corresponding search methods are an effective choice to obtain a sequence from such partial orderings, as detailed in Section 6.4.

As for the problem of where and how to grasp, it is highly conditioned by the particular features of each application. Next, some of such applications are revised, in order of increasing rigidity of the material.

6.1. Cloth handling.

In cloth (un)folding, shape shifts between the original planar extended state of the piece of cloth and a folded state, which can in turn display an aleatory (when the cloth is taken out from a wash bin, for example) or an ordered fold pattern. The constraints influencing the motion of cloth are of two types, as pointed out in [23]: continuous constraints like the material properties of fabric (internal constraints, possibly affected by ambient conditions) and forces like gravity and wind (external constraints), and discontinuous constraints that arise from collisions with other objects. Manipulation constraints —forces applied at specific points in given directions— are the means of controlling the desired folding and unfolding operations. Whereas the continuous constraints are directly considered in the model and its numerical treatment, collisions have to be handled separately. Once detected, an adequate collision response determines the new locations and velocities of the affected points [23, 53, 70].

Continuous constraints and collision detection and response have to be used on a model (Section2.2) to predict the behavior of the cloth when a specific force (manipulation constraint) is exerted on a given point. Such predictions, corrected by current sensorial information, constitute the input of an action selection strategy to determine where the robotic hand should grasp the piece of cloth and how to move with the pinched fabric in order to perform folding and unfolding operations [28]. At a more abstract level, precedence constraints ("don't fold the body before folding the sleeves"), user-defined or arising from geometric considerations, are considered in action sequencing. Pure geometric models plus simplifying assumptions on the physics of garment handling are used in cloth folding [71, 72]

12

Most robotic cloth handling applications up to date use simple models like coarse spring models, the geometric outline of the piece of garment, or no model at all. Contributions include indirect positioning systems [73], unfolding tasks [74–77], folding towels [77], or wiping motions [78].

6.2. Paper folding (origami).

Figure 4 displays typical (so-called *Pureland*) origami folds. One of the faces of the crease can be considered as a fixed *base*, whereas the other one is the rotating *flap*. The number of layers involved in a fold characterizes one-layer (only one flap), some-layers, or all-layers (at once) folds.
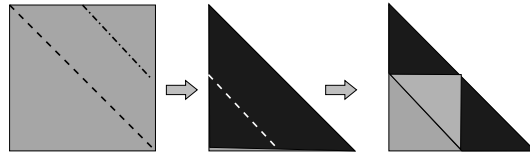
Figure 4: Crease pattern and the corresponding folds. The standard convention of noting valleys by dashed and mountains by dash-dot lines is followed. A simple fold (a symmetrical fold that leaves the same amount of paper on both sides of the crease) is followed by a book fold (±180° folds, simple folds are a subcategory).

The young field of *computational origami* deals with the *origami folding* problem, i.e., to determine whether a particular crease pattern on the unfolded paper sheet can be folded into anything. Whereas *local foldability* (checking the consistency of the mountain-valley assignment to the crease pattern around a single vertex) is non-trivial but can be solved in linear time [79], global foldability —computing the overlap order of the crease faces that fold to a common portion of the plane— is NP-hard in general [67]. Simpler cases are map folding [80] and foldability of paper bags [81].

A robotic origami folding system is described in [82]. The type of origami made by the system is of the *flat* kind, i.e., all facets lie on a single plane (theoretically, actually they are stacked in a given order). See [82, 83] for necessary conditions for flat-foldability. A planner determines the sequence of book folds given the crease pattern on the sheet and the desired stacking of facets. Here, the robotic arm just performs pick-and-place (or machine feeding) operations, in a clamp that performs the actual fold. The dexterous two-handed manipulation required for folding paper as humans do is still an open issue as for a robotic implementation.

In *Nanostructure Origami* [84], a 2D substrate is patterned, creating creases on it and subsequently folding the hinges to spatial configurations (Figure 5). The manufacturability of a given 3D nanostructure, whose layout is conditioned by the specifications and requirements, is tested by trying to unfold it to a flat configuration [84]. The dynamics of an *accordion origami* constructed (i.e., folded) by applying the *stress actuation method* on a patterned membrane is also studied.
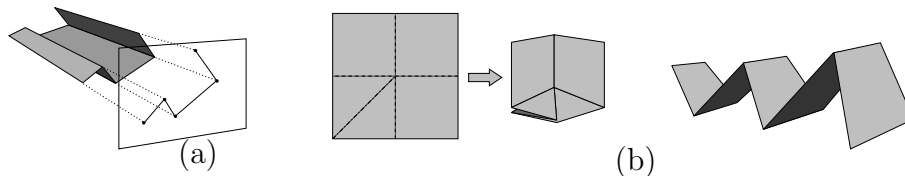

Figure 5: (a) Simple polygonal chains in 2D are related to origami and sheet metal bending as they can be viewed as their projection on a plane which is perpendicular to the direction of the folds. (b) 3D structures obtained by folding. On the left, a corner-like structure. On the right, accordion origami.

6.3. Carton folding.

Carton folding is related to origami, but even more to folding linkages. This has driven the work in [85, 86] to model the carton as an articulated structure (the links are the sides or flaps of the cardboard carton and the folds are rotational joints with bounded rotation angles). Motion planning strategies are applied to characterize the fixtures' layout and the valid paths of such polyarticulated branching manipulators, i.e., the valid folding sequences for the carton, avoiding self-collisions. The authors in [85] point at the potential use of their techniques to sheet metal bending and to the design of 3D MEMS structures from 2D hinged elements.

6.4. Sheet metal bending.

Particular features of this application include: bending angles remain fixed once bent, simultaneous bends are in most cases collinear [85, 87], and the sheet metal remains piecewise planar during the bending process [67]. Process planning involves a task which is directly linked to manipulation planning: determining a feasible (alternatively sub-optimal or even optimal) sequence of bending operations. These operations are performed on a press-brake by positioning the part on the die, positioning the punch on the part, pressing, removing the punch and retracting the part (Figure 6).
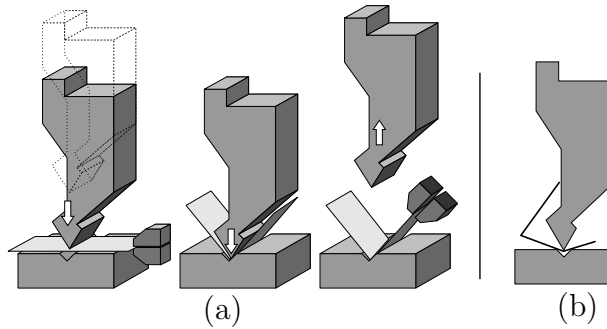
Figure 6: (a) Bending operations, as described in the text, from left to right. During the first and last operations, manipulation of the part has to take possible collisions with the press into account (where the shape of the part is different in the two cases). (b) During bending, a collision occurs between the part and the punch.

The input to the planner is the sheet metal blank, already cut and with known bend lines, as well as a description of the final 3D form of the piece. The output is an ordered set of bending operations. Variables are the used tooling stages (i.e., the different length sets of contiguous punches and dies) and the orientation of the piece when placed on the die (Figures 7 and 8).
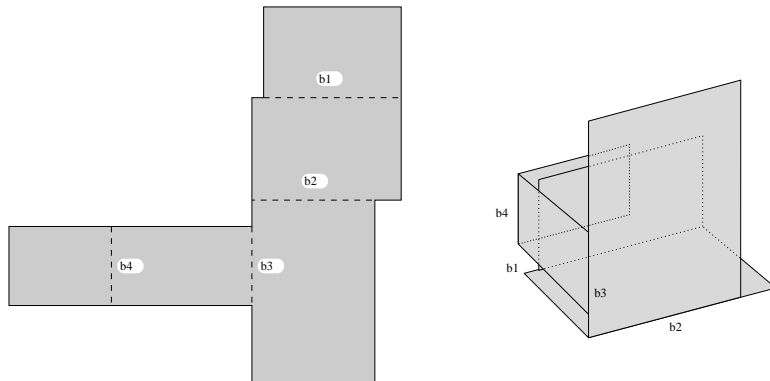

Figure 7: Sheet metal blank with bend lines, and final product. Collision constraints impose partial orderings on the bending sequence: for example, b1 cannot be bent after b3 and b4.

Feasibility constraints include collision avoidance and tolerance specifications. The detection and handling of collisions that may arise during the bending operations (Figure 6) is surveyed in [7], and includes sweep-volume collision verification [88, 89], tuning of tool selection [90, 91], or pruning based on grouping of operations [92]. Constraints on admissible tolerances
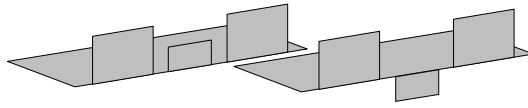
15

Figure 8: With an adequate arrangement of punches and dies, all three bendings can be performed in a single step for the part depicted above, whereas for the one shown on the bottom requires an additional reorientation of the part.

may reject specific sequences of bending operations due to the accumulation of errors [90, 91]. Optimality constraints, on the other hand, arise from ergonomic considerations, applicable to robotic manipulation as well, like the number of workpiece orientation changes [93], combined rotations [90], fixed penalties for each rotation direction [94], or combined penalties for stability and manipulability attributes [95]. Specific manipulability by robots has been addressed in [96] (constraints are the best grasping positions and repositioning timings) and in [87, 97].

The most comon way of computing the sequence (directly or in a folding-by-unfolding fashion) consists in performing a search in a state graph applying the standard heuristic A* algorithm [87, 94, 95, 97, 98] or branch-and-bound strategies [95, 99]. Constraint programming arises also as a natural computation paradigm [100].

The linear counterpart of sheet metal bending is **wire or tube bending**. It is closely related to linkage folding/unfolding, and [101] provide an efficient algorithm do determine if a "linkage" can be straightened including the restrictions that each "joint" can be altered at most once, avoiding collisions and self-collisions, and folding must be done sequentially from one or both ends of the linkage.

7. Topological modifications

In this section we will address topology changes on linear objects. Modifying the topology of higher-dimensional objects means to perform cut and/or glue operations, and have only been treated in a non-robotic surgery context, in simulations [6]. Topological constraints determine the appropriate sequence of operations formulated at an abstract level, and geometric and manipulation constraints can be used afterwards to refine and implement the plan.

7.1. Knotting/unknotting.

Early fixed-sequence robotic knot-tying systems, like [102], stressed hand-eye coordination control issues. Planning how to manipulate a rope or thread in order to untangle a knot, or for tying a knot on itself or around another object means not only to be able to reproduce the physical behavior of the rope realistically, like the simulations in [39, 40], but also to address topological questions arising from knot theory.

The sequence of operations that are necessary to obtain a certain knot can be computed following a tying-by-unknotting approach, knotting would then consist in reversing the obtained sequence and the operations. The search space is a graph whose nodes are crossing states (layered by number of crossings) and the edges correspond to unknotting operations (labelled by operation type), see Figure 9. Figure 10 displays some actions for uncrossing operations (see [103, 104] for a complete characterization).
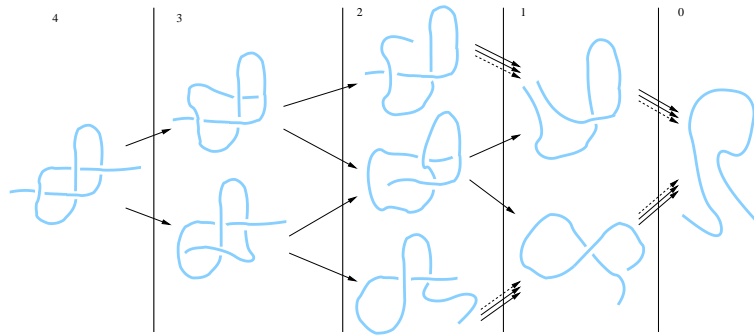

Figure 9: Search space for knotting/unknotting the eight knot, considering only uncrossing operations, after the formalism in [103–105] (where the interested reader can find the graph corresponding to the slip knot displayed at the center of Figure 11). Solid arrows correspond to type-IV uncrossing operations whereas dashed arrows represent type-I uncrossing operations. Some of them appear twice for the different choices of manipulated segments. The columns stand for the number of crossings.

In [103] the authors show that a sequence consisting exclusively of operations involving one endpoint of the rope can always be found, which can be performed by one-handed manipulation (experiments with a robot with three translational and one rotational degrees of freedom are reported). In [105] a procedure for checking the tightenability of knots (Figure 11) is provided.

This latter issue is also addressed in [43], who extend the definition of crossing configuration to include also rigid obstacles around which knots are
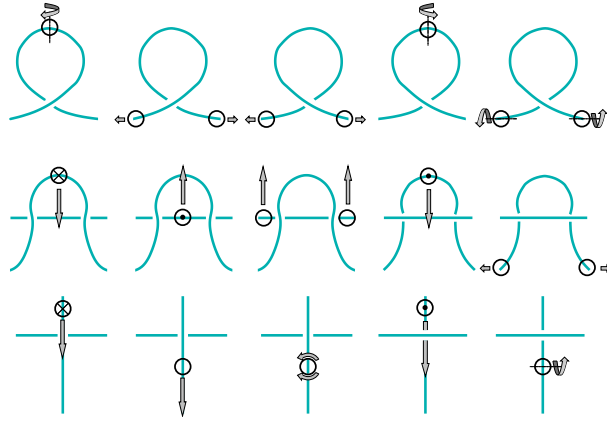
17

Figure 10: Some actions for basic uncrossing operations, after [104] (type-I, -II, and -IV in the first, second and third rows, respectively). Circles are located on grasping points, moving directions are shown with arrows, and approaching directions are represented on the circles with a cross (from the front) or with a dot (from behind). Empty circles mean that the approaching direction is not relevant.
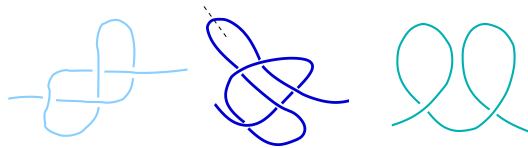

Figure 11: From left to right, a completely tightenable knot (it can be tightly tied by pulling at the endpoints), a partially tightenable knot (which requires an additional segment to be pulled away, this can be detected by cutting at the dashed line: one of the resulting parts is completely tightenable), and an untightenable knot.

tied (or to avoid unwanted loops around them). Randomized motion planning algorithms are adapted to include topological consistency tests when checking feasible configurations [43, 106].

Knots involving several strings and braids are still an unexplored field as for manipulation planning. Braid groups and braid theory may be an enlightening formal framework for this research.

8. Assembly

In most cases, the goal is not to attain a given final shape of the object, but rather to reach specified configurations of its endpoints: a pipe connecting a pump and a vessel, a wire connecting a power supply and a circuit board, a linear spring exerting a given force on its extremes. In the first two examples the exact shape of the object is unimportant (as long as there are no kinks in the hose) whereas in the latter it depends on the required precision of the exerted forces. Sometimes the goal is to force the deformable object through a hole or to lay it along a guiding groove, but again the shape is not precisely defined.

Virtual cables have been modelled by a sequence of cylindrical links connected by ball joints and spiral springs at the joints, for virtual reality applications [107, 108]. Deformable models describing the behavior of these objects have been widely used [109–111] (see also [10] for general three-dimensional objects). These open-loop algorithms are useful for feasibility studies, but are prone to failure in real settings due to uncertainties. In [2] different possible contact states between a linear deformable object and a rigid polyhedral body are identified, and the feasible transitions between these states are listed, as displayed in Figure 12 (this formalism is further extended in [112], characterizing contact states by their stability and defining contact state transition classes).

This provides a robust framework for describing assembly tasks. In later papers, these transitions are identified on sensor-based evidence, both on a visual [113] and force basis [114]. Extensions to non-polyhedral environments are still an open issue. A variant of the probabilistic roadmap method, performing constrained sampling near the contact space, is used in [115] for the problem of cable route planning. The generated milestones are close to features like the corners of a building.
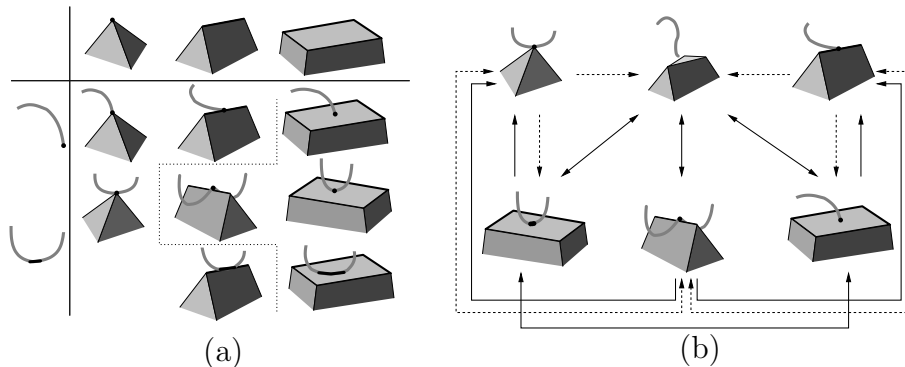
Figure 12: Contact states between the features of a DLO (vertex and edge) and the features of a polyhedron (vertex, edge, face), after [2]. (a) The dotted line separates stable (on the right) from unstable states. Punctual and linear edge/edge and edge/face contacts are distinguished. (b) In the state transition graph, the vertex/vertex and linear edge/edge contacts are not considered, as they are unlikely to occur as initial contact states, and the punctual and linear edge-face contacts are considered together. The non-contact state is represented (top center). Solid links indicate reversible transitions: the outcome can be ensured by a controlled manipulation. Transitions starting at instable states have several possible stable successors and are shown in dashed lines.

9. Conclusions

The suitability of different model types is related to the type of deformation: curves and surfaces are appropriate for elastic deformations, mass-spring meshes and particle systems for elastic and flexible deformations, and polyarticulated analogues for plastic deformations. As for planning strategies, path planning lies at the lower abstraction level in our categorization, and random sampling methods have been successfully adapted to cope with this problem. There are still variants of these methods to be tested for suitability in our context. The main research challenge consists, however, in efficiently biasing this basic search procedures towards the significative parts of the configuration space, as has been described for the latter three manipulation goal categories. There, plans are generated that prescribe a sequence of motions and actions at a level of certain abstraction, what in [104, 105] is called a *qualitative manipulation plan*. There are still open research issues as for effective formulations of qualitative plans, like the definition of "discrete" states for flexible material (e.g. straightened, folded, squashed, crumbled, etc.) that has to cope with a high degree of fuzzyness. Kinodynamic planning [116, 117], on the other hand, operates directly in the space

of dynamic models and its use in manipulation planning of deformable objects is still quite unexplored. Possible integration directions between sensor- and model-based manipulation, like learning basic manipulation skills or uncertainty handling, also deserve further exploration.

Along this survey, a few specific suggestions on punctual open issues have been done. Besides, other open questions include bimanual manipulation of deformable objects (some pioneering work has been done in [43]), which is clearly an issue of concern for humanoid robots performing everyday tasks, and dexterous handling of deformable objects, like multifingered manipulation of planar deformable objects using a vision-based control law [118], or one-handed knot-tying using tactile sensors [119].

References

[1] P. Jiménez, Short survey on manipulation planning for rigid objects, http://www-iri.upc.es/people/jimenez/manplanrigid.html (2008).

[2] A. Remde, D. Henrich, H. Wörn, Manipulating deformable linear objects - contact state transitions and transition conditions, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 3, 1999, pp. 1450–1455.

[3] D. Henrich, H. Wörn (Eds.), Robot Manipulation of Deformable Objects, Advanced Manufacturing Series, Springer Verlag, London, 2000.

[4] M. T. Mason, Progress in nonprehensile manipulation, The Int. Journal of Robotics Research 18 (11) (1999) 1129–1141. doi:10.1177/02783649922067762.

[5] S. Gibson, B. Mirtich, A survey of deformable modeling in computer graphics, Tech. Rep. TR-97-19, Mitsubishi Electric Research Lab, Cambridge(MA) (November 1997).
URL "citeseer.ist.psu.edu/article/gibson97survey.html"

[6] U. Meier, O. López, C. Monserrat, M. C. Juan, M. Alcañiz, Real-time deformable models for surgery simulation: a survey, Computer Methods and Programs in Biomedicine 77 (2005) 183–197, "doi:10.1016/j.physletb.2003.10.071".

[7] J. R. Duflou, J. Váncza, R. Aerens, Computer aided process planning for sheet metal bending: a state of the art, Comput. Ind. 56 (7) (2005) 747–771. doi:http://dx.doi.org/10.1016/j.compind.2005.04.001.

[8] C. Holleman, L. E. Kavraki, J. D. Warren, Planning paths for a flexible surface patch, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 1998, pp. 21–26.

[9] F. Lamiraux, L. E. Kavraki, Path planning for elastic plates under manipulation constraints, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 1999, pp. 151–156.

[10] F. Lamiraux, L. E. Kavraki, Planning paths for elastic objects under manipulation constraints, The Int. Journal of Robotics Research 20 (3) (2001) 188–208. doi:10.1177/02783640122067354.

[11] L. J. Guibas, C. Holleman, L. E. Kavraki, A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 1, 1999, pp. 254–259. doi:10.1109/IROS.1999.813013.

[12] N. Magnenat-Thalmann, P. Volino, From early draping to haute couture models: 20 years of research., The Visual Computer 21 (8-10) (2005) 506–519.

[13] S. Kawabata, The Standardization and Analysis of Hand Evaluation, 2nd Edition, Hand Evaluation and Standardisation Committee (HESC), Textile Machinery Society of Japan, Osaka, Japan, 1980.

[14] J. Hu, Y.-F. Chan, Effect of fabric mechanical properties on drape, Textile Research Journal 68 (1) (1998) 57–64. doi:10.1177/004051759806800107.

[15] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, Elastically deformable models, in: SIGGRAPH '87: Proc. of the 14th

annual conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, 1987, pp. 205–214. doi:http://doi.acm.org/10.1145/37401.37427.

[16] J. Collier, B. Collier, G. O'Toole, S. Sargand, Drape prediction by means of finite elements analysis, Journal of the Textile Institute 82 (1) (1991) 96–107.

[17] D. Baraff, A. Witkin, Large steps in cloth simulation, in: SIGGRAPH '98: Proc. of the 25th annual conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, 1998, pp. 43–54. doi:http://doi.acm.org/10.1145/280814.280821.

[18] J. W. Eischen, S. Deng, T. G. Clapp, Finite-element modeling and control of flexible fabric parts, IEEE Comput. Graph. Appl. 16 (5) (1996) 71–80. doi:http://dx.doi.org/10.1109/38.536277.

[19] S. T. Tan, T. N. Wong, Y. F. Zhao, W. J. Chen, A constrained finite element method for modeling cloth deformation, The Visual Computer 15 (2) (1999) 90–99. doi:10.1007/s003710050164.

[20] D. Breen, D. House, M. Wozny, A particle-based model for simulating the draping behavior of woven cloth, Textile Research Journal 64 (11) (1994) 663–685.

[21] B. Eberhardt, A. Weber, W. Strasser, A fast, flexible, particle-system model for cloth draping, IEEE Comput. Graph. Appl. 16 (5) (1996) 52–59. doi:http://dx.doi.org/10.1109/38.536275.

[22] K.-J. Choi, H.-S. Ko, Stable but responsive cloth, ACM Transactions on Graphics, SIGGRAPH 2002 21 (3) (2002) 604–611.

[23] P. Volino, M.Courchesne, N. Magnenat-Thalmann, Versatile and efficient techniques for simulating cloth and other deformable objects, in: Proc. of Siggraph '95, ACM Press, New York, NY, USA, 1995, pp. 137–144. doi:http://doi.acm.org/10.1145/218380.218432.

[24] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behavior, in: W. A. Davis, P. Prusinkiewicz (Eds.), Graphics Interface '95, Canadian Human-Computer Communications Society,

1995, pp. 147–154.
URL `citeseer.ist.psu.edu/provot96deformation.html`

[25] Z. Wu, C. Au, M. Yuen, Mechanical properties of fabric materials for draping simulation, Int. Journal of Clothing Science and Technology 15 (1) (2003) 56–68. doi:10.1108/09556220310461169.

[26] M. Salzmann, F. Moreno-Noguer, V.Lepetit, P.Fua, Closed-form solution to non-rigid 3d surface detection, in: European Conference on Computer Vision (ECCV), 2008.

[27] F. Moreno-Noguer, M. Salzmann, V.Lepetit, P.Fua, Capturing 3d stretchable surfaces from single images in closed form, in: Conference in Computer Vision and Pattern Recognition (CVPR), 2009.

[28] S. Cuen, J. Andrade-Cetto, C. Torras, Action selection for robotic manipulation of deformable objects, in: Proc. of the ESF-JSPS Conf. on Experimental Cognitive Robotics, 2008.

[29] E. D. Demaine, J. O'Rourke, A survey of folding and unfolding in computational geometry, in: J. E. Goodman, J. Pach, E. Welzl (Eds.), Combinatorial and Computational Geometry, Vol. 52 of Mathematical Sciences Research Institute Publications, Cambridge University Press, 2005, pp. 167–211.

[30] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, A. M. Okamura, Steering flexible needles under markov motion uncertainty, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2005, pp. 120–125.

[31] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, A. M. Okamura, Nonholonomic modeling of needle steering, Int. Journal of Robotics Research 25 (5-6) (2006) 509–525.

[32] M. Moll, L. E. Kavraki, Path planning for minimal energy curves of constant length, in: Proc. of The IEEE Int. Conf. on Robotics and Automation, IEEE Press, New Orleans, LA, 2004, pp. 2826–2831.

[33] M. Moll, L. Kavraki, Path planning for deformable linear objects, IEEE Transactions on Robotics 22 (4) (2006) 625–636. doi:10.1109/TRO.2006.878933.

[34] R. Gayle, P. Segars, M. C. Lin, D. Manocha, Path planning for deformable robots in complex environments, in: Robotics: Science and Systems 2005, 2005, pp. 225–232.

[35] F. Wang, E. Burdet, V. Ronald, H. Bleuler, Knot-tying with Visual and Force Feedback for VR Laparoscopic Training, in: 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE-EMBS), 2005.

[36] I.-K. Jeong, I. Lee, An oriented particles and generalized spring model for fast prototyping deformable objects, in: Eurographics 2004, Short Presentations, 2004, pp. 41–44.

[37] D. K. Pai, Strands: Interactive simulation of thin solids using cosserat models, in: Proc. of the Eurographics 2002, Comput. Graph. Forum, Vol. 21, 2002, pp. 347–352.

[38] M. LeDuc, S. Payandeh, J. Dill, Toward modeling of a suturing task, in: Graphics Interface, 2003, pp. 273–279.

[39] J. Phillips, A. Ladd, L. Kavraki, Simulated knot tying, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2002, pp. 841–846.

[40] J. Brown, J.-C. Latombe, K. Montgomery, Real-time knot-tying simulation, Vis. Comput. 20 (2) (2004) 165–179. doi:http://dx.doi.org/10.1007/s00371-003-0226-y.

[41] J. E. Hopcroft, J. K. Kearney, D. B. Krafft, A case study of flexible object manipulation, Int. J. Rob. Res. 10 (1) (1991) 41–50. doi:http://dx.doi.org/10.1177/027836499101000105.

[42] J. Wolter, E. Kroll, Toward assembly sequence planning with flexible parts, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 1996, pp. 1517–1524.

[43] M. Saha, P. Isto, Manipulation planning for deformable linear objects, IEEE Transactions on Robotics 23 (6) (2007) 1141–1150. doi:http://dx.doi.org/10.1109/TRO.2007.907486.

[44] P. Jiménez, F. Thomas, C. Torras, Collision detection: a survey, Computers and Graphics 25 (2) (2001) 269–285.

[45] G. van der Bergen, Efficient collision detection of complex deformable models using aabb trees, Journal of Graphic Tools 2 (4) (1997) 1–13.

[46] F. Ganovelli, J. Dingliana, C. O'Sullivan, Buckettree: Improving collision detection between deformable objects, in: In SCCG2000 Spring Conf. on Comp. Graphics, 2000, pp. 156–163.

[47] G. Zachmann, Minimal hierarchical collision detection, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST 2002, 2002, pp. 121–128.

[48] M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets, M. Gross, Optimized spatial hashing for collision detection of deformable objects, Proceedings of Vision, Modeling, Visualization (VMV 2003) (2003) 47–54.
URL `citeseer.ist.psu.edu/teschner03optimized.html`

[49] W. Chen, H. Wan, H. Zhang, H. Bao, Q. Peng, Interactive collision detection for complex and deformable models using programmable graphics hardware, in: VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology, ACM, New York, NY, USA, 2004, pp. 10–15. doi:http://doi.acm.org/10.1145/1077534.1077539.

[50] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Straßer, P. Volino, Collision detection for deformable objects, Comput. Graph. Forum 24 (1) (2005) 61–81.

[51] P. Volino, N. Magnenat-Thalmann, Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity, in: Proc. of Eurographics '94, Computer Graphics Forum, Vol. 13(3), 1994, pp. 155–166.

[52] G. Baciu, W. S.-K. Wong, Image-based collision detection for deformable cloth models, IEEE Transactions on Visualization and Computer Graphics 10 (6) (2004) 649–663. doi:http://dx.doi.org/10.1109/TVCG.2004.44.

[53] P. Volino, N. Magnenat-Thalmann, Interactive cloth simulation: Problems and solutions, JWS97-B, Geneva, Switzerland (1997).

[54] J. Mezger, S. Kimmerle, O. Etzmuß, Hierarchical Techniques in Collision Detection for Cloth Animation, Journal of WSCG 11 (2) (2003) 322–329.

[55] T. Larsson, T. Akenine-Möller, A dynamic bounding volume hierarchy for generalized collision detection, Computers & Graphics 30 (3) (2006) 451–460.
URL http://www.mrtc.mdh.se/index.php?choice=publications&id=1122

[56] D. Baraff, A. Witkin, M. Kass, Untangling cloth, in: SIGGRAPH '03: ACM SIGGRAPH 2003 Papers, ACM Press, New York, NY, USA, 2003, pp. 862–870. doi:http://doi.acm.org/10.1145/1201775.882357.

[57] P. Volino, N. Magnenat-Thalmann, Resolving surface collisions through intersection contour minimization, ACM Trans. Graph. 25 (3) (2006) 1154–1159. doi:http://doi.acm.org/10.1145/1141911.1142007.

[58] P. Agarwal, L. Guibas, A. Nguyen, D. Russel, L. Zhang, Collision detection for deforming necklaces, Comput. Geom. Theory Appl. 28 (2-3) (2004) 137–163. doi:http://dx.doi.org/10.1016/j.comgeo.2004.03.008.

[59] E. Anshelevich, S. Owens, F. Lamiraux, L. E. Kavraki, Deformable volumes in path planning applications, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2000, pp. 2290–2295.

[60] O. B. Bayazit, J.-M. Lien, N. M. Amato, Probabilistic roadmap motion planning for deformable objects, in: Proc. IEEE Int. Conf. on Robotics and Automation, IEEE, 2002, pp. 2126–2135.

[61] R. Gayle, M. Lin, D. Manocha, Constraint based motion planning of deformable robots, in: Proc. of The IEEE Int. Conf. on Robotics and Automation, IEEE Press, 2005, pp. 1046–1053.

[62] J. J. Kuffner, S. M. LaValle, Rrt-connect: An efficient approach to single-query path planning, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 00, 2000, pp. 995–1001.

[63] D. Hsu, J.-C. Latombe, R. Motwani, Path planning in expansive configuration spaces, International Journal Computational Geometry & Applications 4 (1999) 495–512.

[64] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, L. E. Kavraki, Sampling-based roadmap of trees for parallel motion planning, IEEE Transactions on Robotics 21 (4) (2005) 597–608.

[65] R. Alterovitz, K. Goldberg, A. M. Okamura, Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2005, pp. 1652–1657.

[66] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, G. S. Chirikjian, Diffusion-based motion planning for a nonholonomic flexible needle model, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2005, pp. 4611–4616.

[67] E. D. Demaine, J. O'Rourke, Geometric Folding Algorithms: Linkages, Origami, Polyhedra, Cambridge University Press, 2007.

[68] R. J. Lang, A computational algorithm for origami design, in: SCG '96: Proc. of the twelfth annual symposium on Computational geometry, ACM Press, New York, NY, USA, 1996, pp. 98–105. doi:http://doi.acm.org/10.1145/237218.237249.

[69] R. J. Lang, Origami Design Secrets: Mathematical Methods for an Ancient Art, A. K. Peters, Ltd., 2003.

[70] P. Volino, N. Magnenat-Thalmann, Implementing fast cloth simulation with collision response, in: CGI '00: Proc. of the Int. Conf. on Computer Graphics, IEEE Computer Society, Washington, DC, USA, 2000, p. 257.

[71] H. Liu, H. Lin, Sequence trajectory generation for garment handling systems, Applied Mathematical Modelling 32 (2008) 1017–1026. doi:10.1016/j.apm.2007.02.026.

[72] J. van den Berg, S. Miller, K. Goldberg, P. Abbeel, Gravity-based robotic cloth folding, in: Proc. of the 9th Int. Workshop on the Algorithmic Foundations of Robotics (WAFR), 2010.

[73] T. Wada, S. Hirai, S. Kawamura, N. Kamiji, Robust manipulation of deformable objects by a simple pid feedback, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, Vol. 1, 2001, pp. 85–90. doi:10.1109/ROBOT.2001.932534.

[74] E. Ono, N. Kita, S. Sakane, Unfolding folded fabric using outline information with vision and touch sensors, Journal of Robotics and Mechatronics 10 (3) (1998) 235–243.

[75] K. Hamajima, M. Kakikura, Planning strategy for task of unfolding clothes, Robotics and Autonomous Systems 32 (2–3) (2000) 145–152. doi:10.1016/S0921-8890(99)00115-3.

[76] M. Kaneko, M. Kakikura, Planning strategy for putting away laundry - isolating and unfolding task, in: Proc. of the 4th IEEE Int. Symposium on Assembly and Task Planning, 2001, pp. 429–434.

[77] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel, Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2010, pp. 2308–2315.

[78] M. Shibata, T. Ota, S. Hirai, Wiping motion for deformable object handling, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2009, pp. 134–139.

[79] M. Bern, B. Hayes, The complexity of flat origami, in: SODA '96: Proc. of the seventh annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996, pp. 175–183.

[80] E. M. Arkin, M. A. Bender, E. D. Demaine, M. L. Demaine, J. S. B. Mitchell, S. Sethia, S. S. Skiena, When can you fold a map?, Comput. Geom. Theory Appl. 29 (1) (2004) 23–46. doi:http://dx.doi.org/10.1016/j.comgeo.2004.03.012.

[81] D. J. Balkcom, E. D. Demaine, M. L. Demaine, Folding paper shopping bags, in: Abstracts from the 14th Annual Fall Workshop on Computational Geometry, Cambridge, Massachusetts, 2004, pp. 14–15.

[82] D. J. Balkcom, M. T. Mason, Introducing robotic origami folding, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2004, pp. 3245–3250.

[83] T. Hull, On the mathematics of flat origamis, in: Congressus Numerantium,100, 1994, pp. 215–224.

[84] P. Stellman, W. Arora, S. Takahashi, E. D. Demaine, G. Barbastathis, Kinematics and dynamics of nanostructured origami, in: Proc. of the ASME Int. Mechanical Engineering Congress and Exposition, Orlando, Florida, 2005, pp. 541–548.

[85] L. Lu, S. Akella, Folding cartons with fixtures: A motion planning approach, IEEE Transactions on Robotics and Automation 16 (4) (2000) 346–356.

[86] J. Liu, J. Dai, An approach to carton-folding trajectory planning using dual robotic fingers, Robotics and Autonomous Systems 42 (2003) 47–63.

[87] S. K. Gupta, D. A. Bourne, K. H. Kim, S. S. Krishnan, Automated process planning for sheet metal bending operations, J. Manufacturing Systems 17 (5) (1998) 338–360.

[88] V. Franke, Automation of tool planning for bent components, in: Proc. of the 3rd Int. Conf. on Sheet Metal, Birmingham, UK, 1995, pp. 35–44.

[89] C. Decubber, P. Schreurs, Werkvoorbereiding voor het plooien van plaatvormige werkstukken, Master's thesis, Department Werktuigkunde, FTW, K.U. Leuven, Belgium (1991).

[90] L. de Vin, A. Streppel, H. Kals, Set-up sequencing procedures for sheet bending, in: Proc. of the 3rd Int. Conf. on Sheet Metal, Birmingham, UK, 1995, pp. 86–95.

[91] L. de Vin, J. de Vries, A. Streppel, E. Klaassen, H. Kals, The generation of bending sequences in a capp system for sheet metal components, Jrnl. of Materials Processing Technology 41 (3) (1994) 331–339.

[92] J. Duflou, H. Dung, J.-P. Kruth, A collision detection algorithm for the verification of sheet metal bending set-ups, in: Proc. of the 6th

Int. Conf. on Sheet Metal, Vol. 2, Enschede, Netherlands, 1998, pp. 209–220.

[93] M. Inui, A. Kinosada, H. Suzuki, F. Kimura, T. Sata, Automatic process planning for sheet metal parts with bending simulation, PED-Intelligent and Integrated Manufacturing Analysis and Synthesis, ASME 25 (1987) 245–258.

[94] M. Shpitalni, D. Saddan, Automatic determination of bending sequences in sheet metal products, in: Annals of the CIRP, Vol. 43/1, 1994, pp. 23–26.

[95] B. Radin, M. Shpitalni, I. Hartman, Two-stage algorithm for rapid determination of the bending sequence in sheet metal products, in: Proc. of the ASME Design Automation Conf., 1996, pp. 1–12.

[96] S. Aomura, A. Koguchi, Optimized bending sequences of sheet metal bending by robot, Robotics and Computer-Integrated Manufacturing 18 (1) (2002) 29–39.

[97] C.-H. Wang, D. Bourne, Design and manufacturing of sheet-metal parts: using features to aid process planning and resolve manufacturability problems, Robotics and Computer-Integrated Manufacturing 13 (3) (1997) 281–294.

[98] J. Duflou, D. V. Oudheusden, J.-P. Kruth, D. Cattrysse, Methods for the sequencing of sheet metal bending operations, Int. Journal of Production Research 37 (14) (1999) 3185–3202.

[99] J. Duflou, J.-P. Kruth, D. V. Oudheusden, Algorithms for the design verification and automatic process planning for bent sheet metal parts, in: Annals of the CIRP, Vol. 48/1, 1999, pp. 405–408.

[100] A. Márkus, J. Váncza, A. Kovács, Constraint-based process planning in sheet metal bending, in: Annals of the CIRP, Vol. 51/1, 2002, pp. 425–428.

[101] E. M. Arkin, S. P. Fekete, J. S. B. Mitchell, An algorithmic study of manufacturing paperclips and other folded structures, THEORY AND APPLICATIONS 25 (2003) 117.

31

[102] H. Inoue, M. Inaba, Hand-eye coordination in rope handling, Robotics Research (1984) 163–174.

[103] H. Wakamatsu, A. Tsumaya, E. Arai, S. Hirai, Planning of one-handed knotting/raveling manipulation of linear objects, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, Vol. 2, 2004, pp. 1719–1725. doi:10.1109/ROBOT.2004.1308072.

[104] H. Wakamatsu, E. Arai, S. Hirai, Knotting/unknotting manipulation of deformable linear objects, The Int. Journal of Robotics Research 25 (4) (2006) 371–395. doi:10.1177/0278364906064819.

[105] H. Wakamatsu, A. Tsumaya, E. Arai, S. Hirai, Manipulation planning for knotting/unknotting and tightly tying of deformable linear objects, in: Proc. of the IEEE Int. Conf. on Robotics and Automation, 2005, pp. 2505–2510.

[106] A. Ladd, L. Kavraki, Motion planning for knot untangling, in: Algorithmic Foundations of Robotics V, Springer Tracts in Advanced Robotics, 2004, pp. 7–23.

[107] E. Hergenröther, P. Dähne, Real-time virtual cables based on kinematic simulation, in: Proc. of the WSCG 2000, 2000.

[108] A. Loock, E. Schömer, A virtual environment for interactive assembly simulation: From rigid bodies to deformable cables, in: Proc. of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI-01): Emergent computing and virtual engineering, 2001, pp. 325–332.

[109] Y. Zheng, R. Pei, C. Chen, Strategies for automatic assembly of deformable objects, in: Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 3, 1991, pp. 2598–2603.

[110] H. Wakamatsu, T. Matsumura, E. Arai, S. Hirai, Dynamic analysis of rod like object deformation towards their dynamic manipulation, in: Proc. of the IEEE/IROS Int. Conf. on Intelligent Robots and Systems, 1997, pp. 196–201.

[111] A. Remde, D. Henrich, S. Karl, H. Wörn, Manipulating deformable linear objects - efficient simulation of the workpiece behavior, in: IASTED Int. Conf. on Robotics and Applications, 1999.

[112] J. Acker, D. Henrich, Manipulation of deformable linear objects: From geometric model towards program generation, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2005, pp. 1541–1547.

[113] F. Abegg, D. Engel, H. Wörn, A robust algorithm for segmenting deformable linear objects from video image sequences, in: Int. Conf. on Pattern Recognition, Vol. 4, 2000, pp. 4756–4759.

[114] A. Remde, D. Henrich, H. Wörn, Manipulating deformable linear objects - force based detection of contact state transitions (2000).

[115] I. Kabul, R. Gayle, M. C. Lin, Cable route planning in complex environments using constrained sampling, in: Proc. of the 2007 ACM symposium on Solid and physical modeling, ACM Press, New York, NY, USA, 2007, pp. 395–402. doi:http://doi.acm.org/10.1145/1236246.1236303.

[116] B. R. Donald, P. G. Xavier, J. Canny, J. Reif, Kinodynamic planning, Journal of the ACM 40 (1993) 1048–1066.

[117] S. M. LaValle, J. J. Kuffner, Randomized kinodynamic planning, The International Journal of Robotics Research 20 (5) (2001) 378–400. doi:10.1177/02783640122067453.

[118] S. Hirai, T. Tsuboi, T. Wada, Robust grasping manipulation of deformable objects, in: Proceedings of the 4th IEEE International Symposium on Assembly and Task Planning, 2001, pp. 411–416.

[119] Y. Yamakawa, A. Namiki, M. Ishikawa, M. Shimojo, One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007, pp. 703–708. doi:10.1109/IROS.2007.4399379.