Project no.: **027657**

Project full title: **Perception, Action & Cognition through Learning of Object-Action Complexes**

Project Acronym: **PACO-PLUS**

Deliverable no.: D7.2.2

Title of the deliverable: Survey on Manipulation Planning of Deformable Objects

Contractual Date of Delivery to the CEC:	**31 January 2008**
Actual Date of Delivery to the CEC:	**31 January 2008**
Organisation name of lead contractor for this deliverable:	**CSIC**
Author(s): Pablo Jiménez and Carme Torras	
Participants(s): CSIC	
Work package contributing to the deliverable:	**WP7**
Nature:	**R/D**
Version:	**1.0**
Total number of pages:	**51**
Start date of project:	**1st Feb. 2006** **Duration: 48 month**

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable contains one scientific publication consisting in a survey on model-based manipulation planning of deformable objects.

Keyword list: manipulation planning, deformable objects, model-based planning.

Table of Contents

INTRODUCTION

This deliverable is related to WP 7.2 - Long-Range Goal Decision Making, which is being developed in conjunction with WP 7.1 - Perception-Learning-Action Loop. For short-range operations, such as those that can be formulated as the optimization of some function (e.g., uncertainty reduction, maximum reachability, stress minimization, etc.), the information-theoretic action-selection mechanism developed in WP 7.1 suffices. However, when the task requires chaining operations to achieve a long-range goal, planning is required. In this context, action selection can be incorporated as one-step planning. When turning to multi-step planning, one has to distinguish between motion planning and task planning. The former has a geometric flavour and the latter a symbolic one. Planning of robot manipulation actions needs both, since it entails performing a task (i.e., fulfilling a long-range goal) which as a subtask requires making motions to reach specific positions with precise orientations.

The task planning part of this deliverable is, thus, closely related to WP 5.1 - Communicative Action Description Language, where the Linear Dynamic Event Calculus (LDEC) representation of OACs, developed in WP 4.3, is extended to attain action recognition, multi-agent planning and communication. In this sense, attention is payed to “affordances”, which are a key ingredient of the symbolic knowledge representation developed within PACO-PLUS.

The delivered “Survey on Manipulation Planning of Deformable Objects”, to be published, captures the evolution and state-of-the-art in model-based manipulation planning. Most planning works up to now deal only with rigid objects, where the state of an object is described by only its pose (position and orientation). However, in the context of the household activities of PACO-PLUS -remarkably, those taking place in a kitchen environment-, there are also non-rigid objects to be manipulated, like cloth, paper foil, or uncountable entities such as flour and sugar, whose state requires much more elaborate descriptions. From the model-based planning viewpoint, deformable objects offer a much richer context for research, since they pose new challenges and more constraints than the mere pick-and-place operations on rigid objects. Here a model describing the behaviour of the manipulated object is crucial. Starting from a coarse dynamic formulation, the model of object behaviour can be adapted by means of learning strategies relying on both sensing and action.

This deliverable is strongly related to WP 6.11 - Combining Learning and Planning. The learning system being developed will generate both primitive rules (one action, with its preconditions and effects) and composite rules (chaining several actions as a reflex to attain a short-range goal), which will constitute the basic building-blocks used by the symbolic task planner to generate long-range goal-directed plans. This general learning scheme is valid to deal with both rigid and deformable objects, the difference lying on the way the object state is represented.

A SURVEY ON MANIPULATION PLANNING OF DEFORMABLE OBJECTS

At the core of a large number of robotic applications, both in industry and outside the shop floor, lies object manipulation. Manipulation planning provides the robot the tools for programming its own movements to accomplish the manipulation task. In the first part of the survey, the classical framework of manipulation planning for rigid objects is described as an extension of well-known robot motion planning strategies. The second and main part is devoted to deformable objects, where manipulation goes beyond mere location changing operations. General considerations about how to model and plan the manipulation of generic volumetric objects are given before illustrating the wealth of applications for specific linear and planar objects. These applications, like assembly of tubes and wires, knotting/unknotting strings or cords, needle steering, cloth handling, origami, carton folding or sheet metal bending, pose specific constraints which have to be considered when planning their manipulation.

ATTACHED PAPER

[A] Jiménez, P. "Survey on Manipulation Planning of Deformable Objects". To be published.

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

1 Introduction

Manipulation lies at the core of a high number of tasks to be performed by a robot. As for industrial applications, it is practically identical with the so-called pick-and-place operations, and constitutes most part of the work involved in assembly or disassembly. Such kind of operations are to be performed a huge number of times, and thus optimality in execution time is crucial, not only by devising an efficient sequencing of the basic tasks, but also at the more elemental manipulation planning level. However, finding optimal or near-optimal paths is a hard problem even in well-known and structured environments. Manufacturing processes do also involve the manipulation of flexible objects, which add an extra degree of complexity, as their own shape is now a variable of the problem. Outside the shop floor, execution time may be not so critical, but the complexity of the environment and of the objects to be manipulated is much higher, due mainly to the richness and unpredictability of possible situations, and to the uncertainty associated to uncontrolled conditions. In particular, when the object to manipulate is deformable, the manipulation actions cover a broad range of activities that cannot be reduced to mere object grasping and displacement operations anymore: knotting a rope or folding a piece of fabric involves considering topology or reasoning about physical processes, respectively, and not just carrying the manipulated objects to stable placements.

Manipulation planning consists in providing an automated system (a robot) the means of programming its own movements in order to accomplish the intended manipulation task. It 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. Different levels may be identified in planning how to manipulate an object, ranging from sensor-based implementations up to the most abstract semantic task formulation. Sensor-based manipulation is the subject of a companion paper, which surveys fixed strategies as well as learning-based approaches. 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. It does also assume that some kind of grasping is involved in manipulation. Thus, nonprehensile manipulation falls out of the scope of this work, see [103, 107, 99] for some relevant work.

This survey is structured in two main parts. The first part is devoted to presenting the classical framework of manipulation planning. The problem is viewed as an extension of the basic motion planning problem [94]. Objects to be manipulated are rigid, and the manipulation task consists in changing their placement while avoiding collisions. This approach reflects the same methodological evolution than motion planning: from complete, exact cell-decomposition methods, in practice suited only for very simple instances of the problem, to sampling based algorithms, able to tackle with more degrees of freedom and more realistic settings. This framework should constitute a formal paradigm for further work involving more complex manipulation, like handling deformable objects. As such objects may also be just transferred from one point to the other, the knowledge of how to tackle this problem with the simpler rigid objects constitutes an unavoidable starting point.

Deformable objects represent new opportunities and do also pose new challenges. Thus, the larger part of the survey (Section 3) is devoted to the manipulation of deformable objects, maybe opening more questions than proposing solutions. To adequately understand all the involved issues, general considerations on how to model volumetric deformable objects and the first algorithms devised to plan the manipulation of this kind of objects are presented. Afterwards, the wealth of possible applications is illustrated with a set of characteristic linear and planar objects. The different goals pursued in these problems pose specific constraints when planning their manipulation. At the same time, an efficient manipulation constitutes also one of the optimality criteria when searching for solutions in such specific settings. The last section is devoted to summarize the main ideas and conclusions to be extracted from the survey.

2 A Manipulation Planning framework for rigid objects

2.1 Exact cell decomposition methods

From its very beginnings, model-based Manipulation Planning reflected the evolution of a closely related discipline, namely Robot Motion Planning. Research stemming from the Computational Geometry community had developed a method for solving the "Piano Movers' Problem" as one of the first exact algorithms for Path Planning [129, 127, 128, 130] (see also [94]). In this series of papers, an upper bound on the time complexity of planning a path in a semi-algebraic free space of given dimension was set for the first time. Similarly, the Manipulation Problem found one of its first formal expressions in terms of a genuine Computational Geometry formulation, in planar environments [156]. Manipulation is viewed there as path planning with movable obstacles: in order to complete a given route (or to arrive to a given goal position), the robot has to displace (to unspecified positions) some of the objects found on its way. Time complexity is also the main issue of concern: NP-hardness is proven by analogy with the 3-SAT instance (the logical operators have their counterparts in gates with movable parts whose position has to be changed by the robot). Although objects are displaced by a robot, this version of the manipulation problem is somehow atypical, in the sense that the accent it put on the robot and its path rather than on the manipulated object. However, in the same reference, a $O(n^2 \log n)$ query time algorithm is sketched (n being the total number of vertices in the environment), after $O(n^3 \log^2 n)$ preprocessing, for a polygonal robot displacing a movable polygon (with a finite set of grasps) to a given goal position amidst fixed obstacles.

The first motion planning algorithms were complete methods, which in practice worked only for very simple, low-dimensional settings. Among them, exact cellular decomposition strategies were quite common, and the algorithm described in [156] fits into this category. Each cell corresponds to the set of collision-free positions for a particular grasp (out of a discrete set of grasps). A reference point for the robot and for the movable object is chosen for defining these cells, and both the robot and the manipulated object are checked for collisions with the environment. Thus, it can be said that the author advanced somehow the idea of combining the configuration space of the robot and the movable object. This formulation was made explicit in the research carried on during the following years [95, 96, 8, 7].

These early works set the standards for the formalization of the Manipulation Problem:

- Manipulation takes place in a composite configuration space ($\mathcal{C}$), which is the cartesian product of the robots' (one or several) and the movable objects' individual configuration spaces.
- In the closure of the free part of this composite space ($cl(\mathcal{C}_{free})$) (legal configurations), two important subsets have to be considered:
  - $\mathcal{C}_{placement}$, where all the movable objects are in valid positions[1], and
  - $\mathcal{C}_{grasp}$, which is the union of the different $\mathcal{C}_{grasp_j}$ corresponding to the configurations where each individual movable object M_j is grasped by one or more robots.
- A *manipulation path* is defined as an alternating sequence of *transit* and *transfer* paths, where the first ones correspond to the motions of the robot without carrying any movable object, while the second ones are those where the movable objects are displaced.
- Transit paths are contained in $\mathcal{C}_{placement}$, whereas the transfer path of object M_j is included in $\mathcal{C}_{grasp_j}$ (Figures 1 and 2 -the latter inspired in Figures 2 and 3 in [134]).
- **Reduction Property** Any two configurations in the same connected component of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ can be connected by a manipulation path.

The last item, the reduction property, provides the clue for planning: compute the connected components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$, and determine their connectivity using transit

[1] These definitions are left intentionally loose in this survey, more precise specifications are given in some references (for example, $\mathcal{C}_{stable}$ is often defined instead of $\mathcal{C}_{placement}$ as the subset of *stable* placements for the movable objects under external forces like gravity).

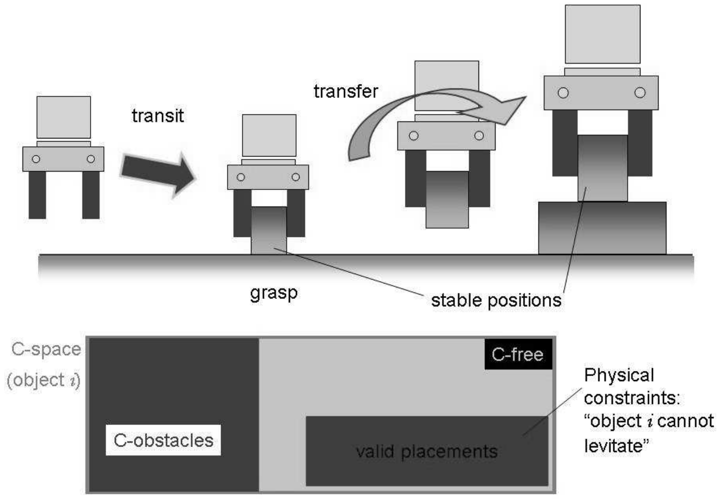

Figure 1: Transit and transfer paths. The set of valid placements of the movable object $\mathcal{C}_{placement}$ is a subset of the collision-free part of its configuration space.

and transfer paths. This is made explicit through the *Manipulation Graph* (*MG*) whose nodes represent these connected components (plus the initial and final configuration of the robot) and whose edges correspond to transfer and transit paths. Manipulation planning consists primarily in a graph search within the Manipulation Graph.

A theoretical method for constructing the connected components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ and their connectivity through transfer and transit paths is provided in [95] for a single or several movable objects: it is based on the $\mathcal{P}$-invariant Collins decomposition of $\mathcal{C}$, where $\mathcal{P}$ is the collection of polynomials that describe $\mathcal{C}_{free}$ and its subsets as semialgebraic sets. Specific algorithms, on the other hand, were developed for very particular settings, like several movable objects with a finite (discrete) set of grasps and placements [8], discs (one robot and one movable object) in a polygonal workspace [96], or one movable (translating, polygonal) object with an infinite set of grasps and placements [7]. These methods have in common that they are only applicable to low-dimensional problems. Involved settings like multi-arm manipulation or handling of three-dimensional objects with multiple re-grasping operations require more powerful algorithms.

2.2 Multi-arm and re-grasp manipulation of rigid objects

Manipulation planning reflects the advent of new paradigms in robot motion planning. For example, the Variational Dynamic Programming method, devised in [23] for computing collision-free paths, is used in [24]. Here, manipulation is considered as a specific instance of the basic path planning problem (with the addition of a formal *grasping constraint*). The Progressive Variational Dynamic Programming method (which is basically a penalty function method), consists in solving a relaxed version of the grasping constraint, and successively tightening this constraint, using the path found at the previous iteration as input.

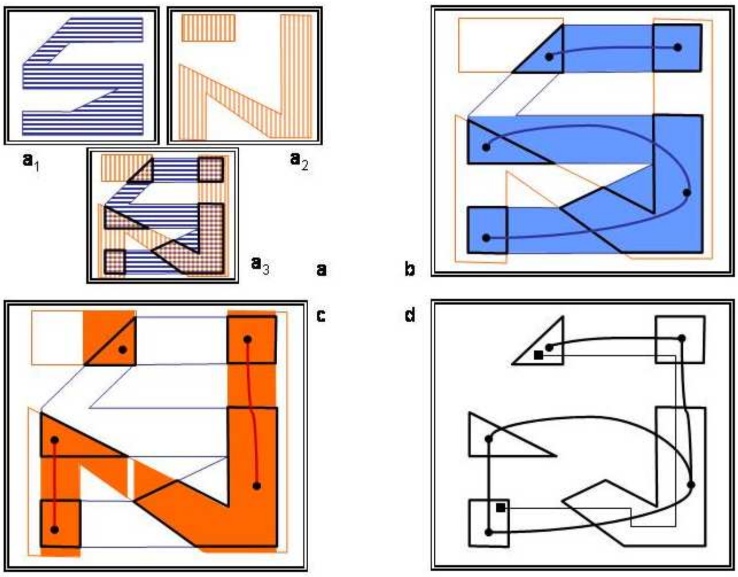

Figure 2: Transit and transfer paths (cont.). (a) The sets representing $\mathcal{C}_{placement}$ (a_1)and $\mathcal{C}_{grasp}$ (a_2), as well as the five components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ (a_3) are displayed. The horizontal pattern in (a_1) represents the *foliation* of $\mathcal{C}_{placement}$, i.e., the partition of this space into lower-dimensional subsets induced by transit paths (each corresponding to a fixed location of the movable object). Also transfer paths induce a foliation in $\mathcal{C}_{grasp}$. This means that the space which is reachable by a transit (resp. transfer) path starting at a configuration inside one of the five components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ is represented by the shadowed regions in (b) (resp. (c)). The connectivity between these components by transit and transfer paths is represented by arcs connecting them. Finally, (d) displays the connectivity of the *Manipulation Graph*, and a possible manipulation path between two given configurations (small squares).

But the most popular methods nowadays are the random sampling based algorithms, and the Randomized Path Planners (RPP), Probabilistic Roadmap Methods (PRM), Rapidly-exploring Random Tree planners (RRT), or Ariadne's Clew Algorithm (ACA), all of them originally devised for solving involved instances of the basic motion planning problem, have their counterparts in the manipulation planning field, as shown in Table 1 (for the original motion planning algorithms, see also the textbooks [94, 36, 97]). These methods stand out for their adequacy for coping with high-dimensional problems.

	Motion Planning	Manipulation Planning
RPP	[25]	[84]
PRM	[81] [135] [38]	[114] [134]
RRT	[86]	[134, 138]
ACA	[29] [108]	[6]

Table 1: Random sampling-based methods for motion and manipulation planning

Difficult manipulation settings addressed by these methods include multi-arm manipulation [84] or manipulation planning for redundant robots [6]. Although the formalism explained in the previous section is kept alive in the mind of the researchers, an explicit and complete characterization of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ is now avoided. Instead, the algorithms try to identify landmarks inside the connected components of this intersection submanifold, ideally representing all of them, which cannot be ensured (if not directly provided by the user, as assumed in [114]), and connect them via transit and transfer paths. Alternatively, the sequence of transfer and transit paths may arise from the computation of a collision-free path of the manipulated object [84].

All these algorithms share a common structure, they operate at two levels: the global level (called task level by some authors) cares about connecting the initial and the goal configurations through a sequence of landmarks in $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$, whereas the local level tries to verify whether these landmarks can actually be linked together. Some algorithms assume that the set of possible grasps is predefined, finite, and discrete ([84, 6, 114]. This allows them to replan a global path when the local planner returns failure for a given grasp configuration. This strategy puts clearly the transfer tasks (and the derived transfer paths) in the foreground, as explicitly done in [84] by assuming that for the subsidiary transit tasks always the corresponding transit paths can be found (a modified RPP is used to compute valid transfer paths). Replanning, which in this context is associated to regrasping, is clearly visualized in the application of ACA to manipulation planning [6] by the generated tree-structure, see Figure 3.

An attempt to lower the replanning burden is presented in [114], whose planning algorithm -called by the authors *Fuzzy PRM*- differs from standard PRM in that edges corresponding to transfer and transit tasks have an associated probability which is an estimation of the chance of this sub-path to be found by a local planner during the query (i.e., local) phase. This work, like the previous ones, does again reproduce the planning structure at both levels, although in this case the learning phase of PRM is assigned to the task level (in a single run, connecting landmarks with the same grasp relation via transfer edges, and landmarks with the same object location with transit edges), whereas

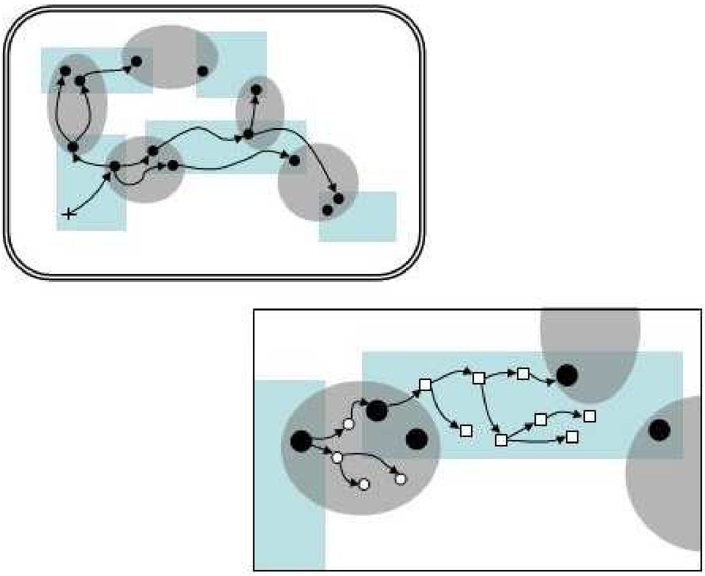

Figure 3: ACA applied to manipulation (from [6]). Up on the left, at the top level, the *EXPLORE* part of ACA (called *E-MANIP*) tries to spread landmarks, which are reachable from the start configuration, in $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ by generating single manipulation paths (a transit followed by a transfer path), whereas the *SEARCH* routine tries to reach the goal from these landmarks via a single transfer path (if it does not succeed, *EXPLORE* places new landmarks starting at the previous ones, generating a tree of paths). Down on the right, the same *EXPLORE-SEARCH* strategy is followed at the lower level of the subproblems generated by *E-MANIP*, i.e., planning the particular transit and transfer paths in $\mathcal{C}_{placement}$ and $\mathcal{C}_{grasp}$ respectively.

local planners run during the query phase (as instances of Fuzzy PRM for the different possible grasps, i.e. for the transfer paths, as well as for the transit paths).

Instead of assuming discrete predefined grasps and to compute valid configurations for the robot grasping the object at a particular position, some authors face the involved problem of characterizing the continuous set of robot grasping configurations. In [134][2] the connected components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$ are computed by determining the parameterization of the set of configurations that satisfy the closure constraints induced by considering the grasping robot and the placed object together as a closed kinematic chain. The latter can be solved with specific versions of PRM like the *Random Loop Generator* developed by the same authors [38]. A *Visibility-PRM* [135] is used to compute small roadmaps for the different connected components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$. Another difference with respect to the object-centered strategies described above is that here a roadmap for the robot is computed first in absence of the movable objects, afterwards it is updated by checking for collisions of the edges with the current position of the movable object (see Figure 4). Bidirectional Rapidly-exploring Random Tree planners [86] are used to locally explore the space around the edges blocked by the movable objects, as shown in Figure 4(c).

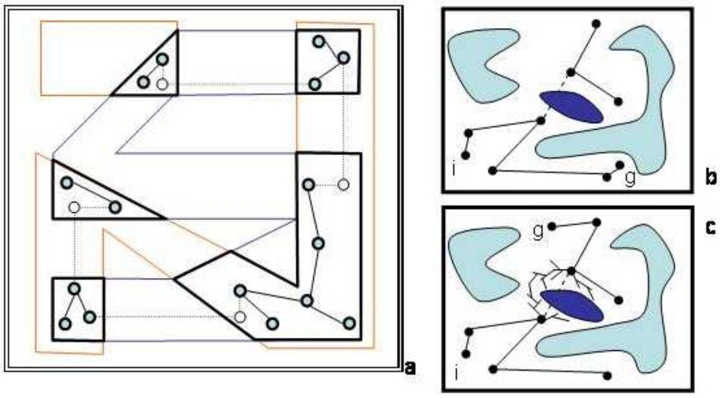

Figure 4: (a) A "Visibility-PRM" is used for computing the small roadmaps inside the connected components of $\mathcal{C}_{placement} \cap \mathcal{C}_{grasp}$. The links of these roadmaps correspond to simultaneous changes of placement and grasp, which are not feasible, but can be reduced to a finite sequence of pure transit and transfer paths. These roadmaps are linked together with simple transit followed by transfer (or vice-versa) paths. (b) and (c) In the configuration space of the robot, a roadmap is computed in absence of the movable objects. Arcs of this pre-computed roadmap may be invalidated due to the presence of the movable objects (darker region). This may have no consequences for a particular query (b), but if it has, RRT-methods are applied locally (c).

A recent contribution [138] does also make use of RRT-planners for the more involved problem of manipulation planning among movable obstacles. Their ResolveSpatialConstraints (RSC) algorithm is, in words of the authors, a reverse-time search that samples future robot actions and constrains the space of prior object displacements.

[2]This paper gathers together the results presented in [132, 133, 126]

2.3 Bridging the gap to task planning

In the aforementioned works, although in some approaches two levels have been distinguished, calling "task level" to the higher one, actually everything has been formulated in geometric terms. However, when properly talking about task planning, an abstract symbolic formulation is generally meant. And complex manipulation tasks may require some kind of symbolic structuring in order to bound the huge search space at the geometric level.

Manipulation planning is viewed in [64] as a geometrical and task (planning) problem. Consequently, a link between the two levels has to be established. This is done, in first instance, through a collection of heuristic rules that constitute "the first level of symbolic control of the purely geometrical PRMs search methods" [64]. In this work, several roadmaps are maintained simultaneously, each corresponding to a particular aspect of the problem (i.e., allowed placements and movements of the object, transit and transfer paths, gripper/object grasp transforms, etc). Basic notions are the "robot composition" (two robots, or a robot and an object, are considered as a single robot, and have an associated roadmap), as well as the aforementioned *reduction property.* The heuristic rules are used to determine which roadmap to expand (that is, where new nodes have to be randomly generated) or where to search next.

This approach has evolved towards the *aSyMov* system [65]. It is basically a planner that selects one of the applicable actions at the symbolic level (basic actions include *goto*, *pick*, and *place*) or roadmap-expansion action, depending on the computed costs and heuristics. This selection is made on a probabilistic basis, i.e., not always the best action is selected. Actions are built from predicates stating preconditions and effects of these actions (basic predicates include statements about the composition of two robots or a robot and an object, or the location of a robot at a symbolic position, for example). The connection to the geometrical part of the planning occurs at the *action validation step*, an "incremental instantiation process of the symbolic positions" (where geometrical constraints may back-propagate to the whole plan under construction, not just apply to the action being validated). The different possible instances (nodes in the roadmap) are kept in *accessibility lists*, which are revisited in case of back-propagation due to invalid instances of selected actions.

Instead of a set of $\mathcal{C}$-spaces where connected collision-free paths have to be found, the lower planning level may be a battery of motion directives implemented in agent-specific behaviors (the broader term "agent" is used here instead of -and subsuming- "robot", as many contributions come from the Computer Graphics field, specifically from Virtual Human simulation). Nonetheless, such elemental procedures are still too specific to be efficiently used at task-level. Thus, an intermediate system like the *Object Specific Reasoner* [98, 47] is necessary for linking the high-level domain-independent task-actions to the completely specified motion directives from the physical (or simulation) world. This system has been developed from the observation that the same task-action, say "pickup", may produce quite different physical actions, graspings in this case, when applied to different objects (or to the same object with different purposes). Objects are grouped into functional categories, which trigger a specific *action outline* for each task-action. All steps of this outline are recursively expanded, until all sub-task-actions are decomposed into motion directives. Possible parameter values are generated, using specific attributes of the agent and the object, and sorted according to the purpose of the action. If the agent's resources are consistent with the object's attributes, the task-action is feasible in the current setting and the instantiated motion directives are transferred to the agent. In perceptual learning, objects are said to afford (or support) specific interactions of the perceiver with the world (see [61, 136, 39], in the last reference a complete survey on the application of the concept of affordances to the control of autonomous robots can be found).

The *Smart Objects* in [79], [80], [1] provide a similar kind of object-centered manipulation paradigm. The geometrical information describing the shapes of objects is extended with semantic information about their behavior when an interaction with an agent occurs. This information includes whole manipulation sequences, which are not limited to grasping, but also consider reaching the objects, looking at them, changing grasps, etc. Grasps are generated semi-automatically in the design phase of the object (in Robotics they could be learned as well), by allowing the designer to select relevant tubular regions for grasping and to specify parameters like wrist position and orientation, thumb configuration, touch tolerance or finger involved in the grasp, while leaving the low-level computations (collision detection and posture search in the configuration space of the hand) to the system. In [63] the system is completed with local perception capabilities that allow the agent to classify the object from its perceived feature set, and to retrieve the actions attached to the corresponding smart object.

The *Task Definition Language* [147] does also aim at bridging the gap between high-level task specifications and the agent's interaction with the virtual environment. The language allows to combine a number of built-in *primitive actions* -like changing the configuration of entities- in a sequential, parallel, or conditional fashion to generate complex tasks. In particular, regarding interactions with objects, grasping can be implemented as a parallel combination of *inverse kinematic* functions moving the fingers concurrently towards destination spots on the object (or rotating the fingers until collision with the object's surface). The final step would be to have the possibility of giving high-level commands in natural language, like the *Parameterized Action Representation* [17, 16]. Although addressed at communicating with virtual humans, i.e., smart avatars in virtual environments, this PAR-based architecture could be inspiring for designing a communication interface with smart embodied agents like robots intended to carry out manipulation tasks. This system is designed to provide a complete description of an action, including the agent that performs it and the list of affected objects, the applicability conditions that must hold, a list of conditional preparatory specifications, the execution steps and the termination conditions.

3 Manipulation of deformable objects

Up to now, physical considerations about the manipulated objects were implicit in (and restricted to) the assumption of feasibility of their allowed placements: $\mathcal{C}_{placement}$ includes only those configurations that are stable under gravitational forces. When the objects to be manipulated are deformable, and their deformations are relevant for the task, other physical constraints must hold, mainly the minimization of the internal elastic energy. Algorithms devised for computing a realistic behavior of deformable objects under manipulation constraints have to cope with an infinite dimensional configuration space.

The type of deformation depends not only on the material properties of the objects, but also on the intensity and duration of the applied forces. A similar characterization is mentioned in [57], this one tries to be more strict:

Elastic The object recovers completely its original shape once the external forces have ceased.

Plastic The elasticity limit has been surpassed and permanent, stable deformations appear on the body of the object. This includes also piercing, breaking or cutting the object.

Flexible Deformations are not permanent and the shape of the object can be altered with very slight effort.

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 5 shows some basic deformation types.

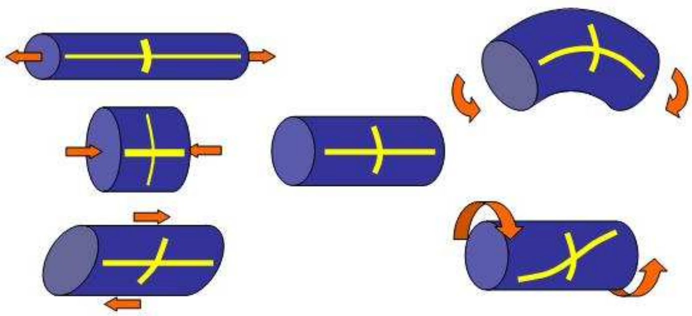

Figure 5: The undeformed cylinder appears in the center of the figure. At the left, tension, compression and shear, and on the right bending and torsion are exemplified.

Conceptually speaking, manipulation planning of deformable objects is a quite different issue from motion planning of deformable robots: in the first case, deformations of a passive object result from its manipulation by rigid arms, whereas in the latter it is an active machine which takes its own deformations into account when computing the intended trajectory. However, the differences blur when considering devices like flexible steerable needles, which are guided by manipulating them at the insertion point. Putting aside the case of flexible manipulator arms, where the main objective is to reach given configurations or to follow a specified path while minimizing vibrations (see, for example, [117, 160, 159, 27] and the excellent survey [53]), in this quick survey the emphasis is put rather on the geometry of the deformable object.

3.1 General deformable volumetric objects

3.1.1 Modelling deformable volumetric objects

Linear and planar objects, like those surveyed below, aren't but abstractions where some simplifications have been undertaken in order to make the related simulations tractable. The fact is that the objects in our world are three-dimensional, and the models that were initially devised for capturing the physics of deformable objects took the three dimensions into account. The most outstanding 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 objects. A first decision when modelling the objects of the planning problem concerns whether to restrict to pure geometry or to consider these underlying physical laws, and, in the latter case, to which extent or degree of accuracy they are captured. In general, the more accurate the description of physics, the higher the computational effort needed. Accurate models, like the Finite Element Method, a classical tool in Engineering, are in fact more appropriate for off-line simulations. Approximate models have been developed in recent years that meet both the requirements of realistic behavior and real time execution. Excellent surveys exist on this subject in the field of Computer Graphics [62] as well as in Surgery Simulation [109], where not only elastic deformations but also incisions and suturing have to be considered. Schematically, existing models can be of the following types:

- **Purely geometric models**.
  - Splines and patches, whose shape is modified by manipulating their control points.
  - Free-form deformations modelled as alterations of the space where the object is embedded in.
- **Physical models**. Physical analogues of elastic behaviors serve as basis for this family of methods, that trade accuracy and fidelity for computational efficiency.
  - Active contours or snakes (deformable splines where a physically-based behavior is induced by associating a parameterized energy to their degree of deformation)
  - Mass-(damper)-spring models
  - Linked volumes
  - Mass-tensor models
- **Continuum models**. They reproduce the spatial variation of material properties according to the laws of Physics, in form of differential equations that are solved by discrete algorithms as the finite difference and the finite element methods.

Also **hybrid models** exist that combine some of the features of these different types. The interested reader is referred to the surveys mentioned above.

3.1.2 Planning with deformable volumetric objects

In the context of manipulation, planning the motion of a deformable object means to determine a collision-free path from an initial to a goal configuration for this object, subject to *manipulation constraints* (i.e., the ways it can be manipulated). The planning algorithm tries to reproduce a realistic behavior of the manipulated object, which means that low-energy equilibrium deformations have to be considered all the time [11]. The shape is conditioned by the way the object is manipulated, which means that for any configuration of the manipulating robot(s) two items have to be tested for: whether the object can actually be deformed this way without surpassing the elasticity limits, and, in the affirmative case, whether the resulting shape does or not collide with the obstacles in the environment. Viewing the deformable object as a robot by itself (in fact, the common approach is to consider the free-flying deformable object without taking the grasping arm(s) into account), the point is that deformations may occur at any point of the object, which translates into infinitely many degrees of freedom. The object could be represented as a polyarticulated device with many degrees of freedom (dof), and a survey on motion planning for such kind of objects can be found in [67]. Some of the applications involving linear and planar objects mentioned below translate directly into a multi-link formulation. However, for objects with continuous deformations, the number of necessary dof may increase to intractable levels, without providing a satisfactory reproduction of the physical behavior [26]. In the tradeoff between physical accuracy and efficiency, some authors opt to use FEM-based methods [140], but geometric or physical models are preferred by large: free-form deformations [26], and more frequently mass-spring models [11, 148, 58]. In the latter, the springs connecting the mass nodes include not only those in a rectangular array, but also diagonal ones.

As for the planning strategy, Probabilistic Roadmap Methods are clearly preferred [11, 26, 58]. In [11] the same approach as in [88] (see Section 3.3.1 below) is followed: first a random manipulation constraint is generated and tested for feasibility (both the plain strain limit and the curvature limit are checked for locally), for valid deformations a given number of randomly generated rigid transformations are checked for collisions with the obstacles. The specification of the manipulation constraints depends on the particular case: an example is shown with a deformable elongated parallelepiped (represented with a lattice of 32x3x3 nodes) where the manipulations constraints consist in the specification of the position and orientation of its both ends. Still following the classic PRM strategy, the algorithm tries to connect the new configurations to the existing roadmap using the local planner, by computing first the necessary rigid transformations and then interpolating between the manipulation constraints to obtain a *deformation path*. The approach followed in [26] differs from the previous one in that feasible paths during roadmap construction may have collisions and in such cases deformations on the object are attempted so as to avoid them. If this is not possible, the corresponding path segment is abandoned. In other words, 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 [58], 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.

3.2 Linear objects

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. We may refer to them generically as *Deformable Linear Objects* (DLO). A classification of the algorithms that deal with this kind of objects is possible attending on the purpose of the manipulation:

3.2.1 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. In the Computer Graphics community, efforts have been devoted to simulate virtual cables, modelled by a sequence of cylindrical links connected by ball joints and spiral springs at the joints, for virtual reality applications [70, 101]. Ad-hoc algorithms exist based on sensory feedback, but, as pointed out in [122], they lack generality. Deformable models describing the behavior of these objects have been widely used [161, 150, 121] (see also [89] 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 [122] 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 6 (this formalism is further extended in [4], characterizing contact states by their stability and defining contact state transition classes).

In this way, a robust framework for describing assembly tasks is provided. In later papers, these transitions are identified on sensor-based evidence, both on a visual [2, 3] and force basis [123]. A variant of the probabilistic roadmap method, performing constrained sampling near the contact space, is used in [78] for the problem of cable route planning. The generated milestones are close to features like the corners of a building. A guiding path is computed on the roadmap and attraction forces are exerted on the first link of the cable (modelled as an articulated body) towards this path. The cable is deformed using adaptive forward dynamics, and collision detection and response ensures that no constraint is violated. The output of this algorithm may serve as a basis for planning a manipulation strategy to attain such a cable layout: during simulation, also additional path constraint forces are computed acting on a subset of links selected for their position with respect to the curved segments of the path followed by the first link. These constraints ensure that the cable keeps its layout along the walls.

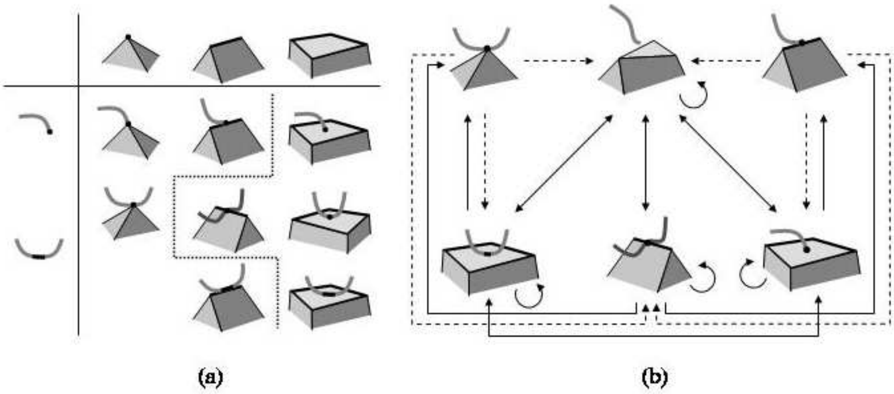

Figure 6: Contact states between the features of a DLO (vertex and edge) and the features of a polyhedron (vertex, edge, face), after [122]. (a) The dotted line separates stable (on the right) from unstable states. Punctual and linear edge/edge and edge/face contacts are distinguished. (b) In the state 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.

3.2.2 Knotting/unknotting

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 [32, 116], but also to address topological questions arising from knot theory. Knot tying is presented as a case study of flexible object manipulation in [72]: together with a graph representation of knots, a knot-tying grammar is described, based on parametric motion commands for the robot. *Tying knots* is also one of the common manipulation operations on strings enumerated in [157], together with *winding* and the elementary operations of *pushing*, *pulling* (with *folding* and *pushing through a hole* as particular instances) and *forming loops*. All these operations and their interdependencies are discussed, not only from the assembly planning perspective, which is the aim of this reference, but also regarding manipulation issues. Knots, whether involving one or various strings, are explicitly represented in the *feature-relationship diagram* of the ropes, just by a label in the form of a dashed edge between the affected nodes, or alternatively including a representation of the interior structure of the knot. Reidemeister moves from knot theory [5] inspire the crossing state changing operations on ropes in [151, 152, 149] (Figure 7).

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 (thus, each crossing point appears twice, see Figure 8). Each part of the rope between two consecutive crossings (or a crossing and an endpoint or extreme of the rope) is called a *segment*.

The sequence of operations that are necessary to obtain a certain knot can be obtained following a tying-by-unknotting approach: the goal state is successively changed with

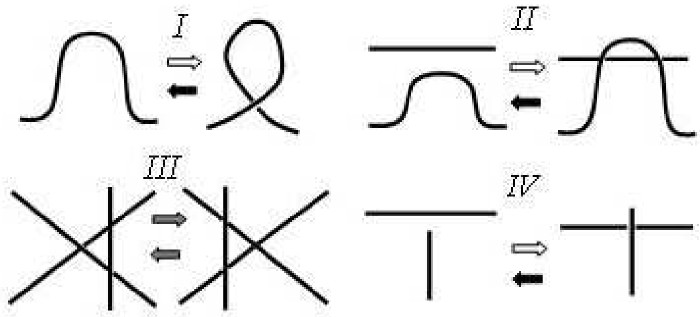

Figure 7: Basic operations in [151, 152, 149]. Type-I, -II, and -IV are crossing (from left to right) and uncrossing (from right to left) operations, whereas type-III are arranging operations (they do not change the number of crossings but their sequence).

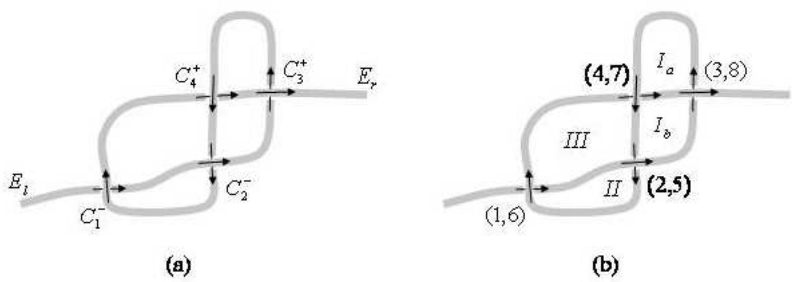

Figure 8: (a) Sequence of crossings representing the state of the rope: $E_l - C_1^{l-} - C_2^{u-} - C_3^{l+} - C_4^{u+} - C_2^{l-} - C_1^{u-} - C_4^{l+} - C_3^{u+} - E_r$. The first two crossings are *left-handed helical* (labelled with a minus sign) and the other two are *right-handed helical* (plus label). The labels u and l refer to the upper and lower part of the crossing, respectively. (b) An alternative representation, used in [125]. 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 in Figure 9), note that the last step pierces loop I in two loops I_a and I_b.

unknotting operations, until the original untangled state is obtained (knotting would then simply consist in reversing the 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). Figure 9 shows an example of such a graph.

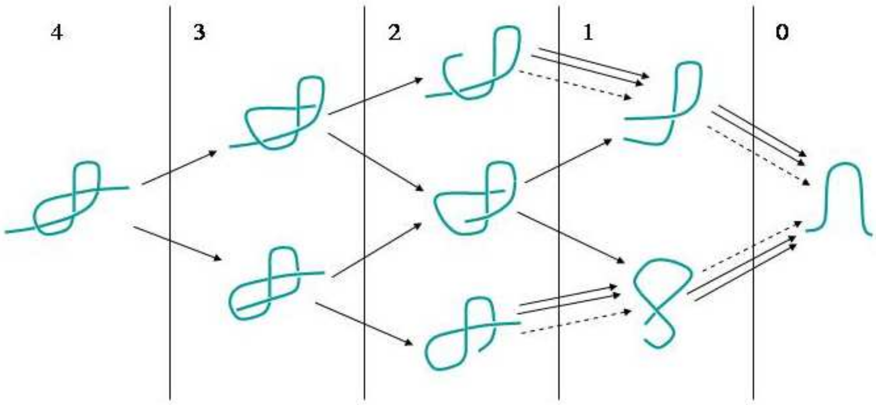

Figure 9: Search space for knotting/unknotting the eight knot, considering only uncrossing operations, after the formalism in [151, 152, 149] (where the interested reader can find the graph corresponding to the slip knot displayed at the center of Figure 11). Solid arrows correspond to type-IV uncrossing 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.

The authors provide furthermore the means to determine the grasping points (characterized in 17 *grasping patterns* [151]) and necessary moving (one of four types of translations and rotations) as well as approaching directions (from the front, from behind, or from either one). The possible combinations of these parameters lead to a set of 34 *actions* for uncrossing operations, 46 for crossing, and 16 for arranging [149]. Figure 10 displays some actions for uncrossing operations.

In [151] 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 [152] a procedure for checking the tightenability of knots is provided (see Figure 11 for the concept of tightenable knots).

This latter issue is also addressed in [125], who extend the definition of crossing configuration to include also rigid obstacles around which knots are tied (or to avoid unwanted loops around them). A single-query probabilistic roadmap is constructed, where the random configurations of the rope are checked not only for collision but also for topological consistency. The possible use of *needles* guiding the rope to form loops is also considered. Randomized motion planning techniques are also used in [87]: energy minimization is combined with randomized tree-based planning for knot untangling, viewed as a high-dimensional planning problem in a reparameterizable configuration space.

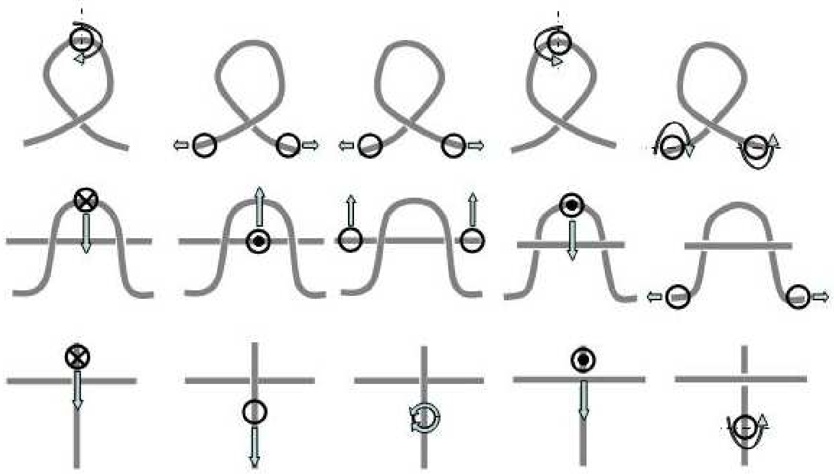

Figure 10: Some actions for uncrossing operations, after [149]. Circles are located on grasping points, moving directions are shown with arrows, and approaching directions are represented on the circles with a cross (from the front) or with a dot (from behind). Empty circles mean that the approaching direction is not relevant.

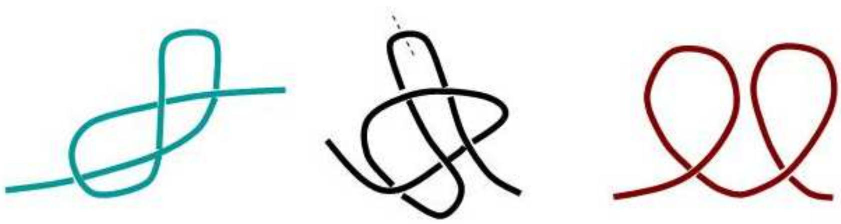

Figure 11: From left to right, a completely tightenable knot (it can be tightly tied by pulling at the endpoints), a partially tightenable knot (which requires an additional segment to be pulled away, this can be detected by cutting at the dashed line: one of the resulting parts is completely tightenable),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

3.2.3 Steering

In some applications like suturing or endoscopic manipulation, the DLO has to follow a specific path, which is closely related to the configurations it attains while minimizing its internal energy and subject to the constraints imposed by its environment. Minimal strain curves in 3D, not subject to stretch (i.e., of constant length), have been considered as valuable models for path planning of DLOs, as for such minimal energy configurations dynamics can be ignored and they are easier to execute [113, 111, 112]. Path planning consists, then, in determining all minimal energy configurations between the start and the goal curves. Geometric constraints (position and tangent) are defined for the endpoints and an arbitrary number of intermediate control points, and the shape of the curve is computed from the relationship between the internal energy and the torsion and curvature of the curve. This energy optimization (using a general purpose constrained optimization technique, which has a better tradeoff between accuracy and efficiency than Lagrangian dynamics or random sampling of the null space of the Jacobian) is performed on a curve defined by ten parameters (for each two consecutive control points, as each segment of a minimal energy curve has minimal energy) [113], as well as for variable resolution curves modelled by a subdivision scheme [111, 112]. Steering flexible needles with bevel tips in soft tissue while avoiding obstacles 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 [154], 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 (Figure 12).

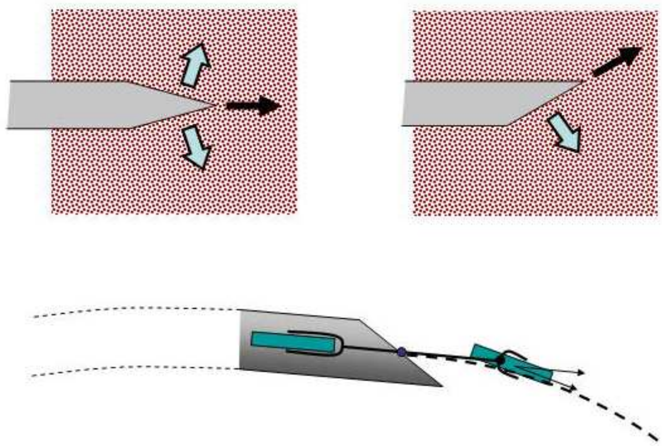

Figure 12: While a symmetric tip (top left) is driven in the initial insertion direction, the reaction forces acting on the asymmetric bevel tip (top right) drive the needle in a curved trajectory. At the bottom, the non-holonomic bicycle model used in [154] (generalized to 6 DOF).

This steering is treated as a nonholonomic motion planning problem for a Dubins car with no reversal in [10]. Planning is performed on a 2D imaging plane, and direction changes may only happen at discrete control points separated by an insertion distance δ (i.e., at each such point the needle is inserted a distance of δ without rotation, or the bevel is rotated 180^0 and inserted by δ). Costs are attached to the insertion depth and to the number of rotations, prohibitive costs preventing collisions with obstacles or leaving the workspace. Transitions between the discrete states of the system conform a Markov Decision Process, the transition probabilities depending on the uncertainty of the needle motion. Thus, minimizing the total expected cost of inserting the needle to a target position is a stochastic shortest path problem optimally solved by using infinite horizon dynamic programming. In [9] the control variables include also the location of the insertion point and insertion angle, and emphasis is put on modelling the soft tissue deformations and simulating the friction along the needle shaft. The 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.

In [115] the planning of the needle path extends to the 3D space (not restricted to the imaging plane) of an isotropic tissue without obstacles (Figure 13). Thus, both linear and angular insertion velocities are variables in the steering model.

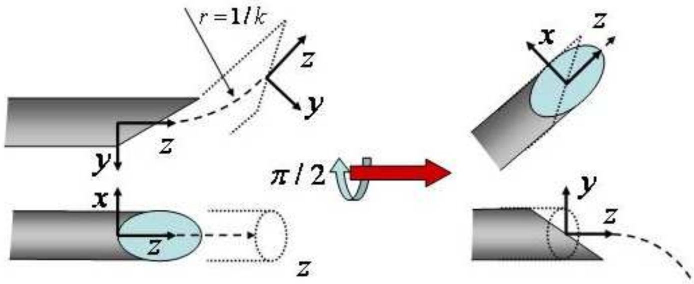

Figure 13: Due to torsional rigidity, the angular velocity at the insertion point causes the tip to rotate and thus to change the direction of progress. A 3D linear trajectory results for the needle. Two views are shown: a side view on top, and a view from below, before and after a given linear displacement and an axial rotation of $\pi/2$.

This nonholonomic trajectory planning problem is accomplished by diffusion-based motion planning on the special Euclidean group SE(3). A flexible catheter for liver chemoembolization that is introduced and guided along blood vessels up to its target is the inspiring application for the DLO motion planning algorithm in [59]. The robot, i.e. the catheter, is represented with a mass-spring model (also [11] use this kind of model for path planning of deformable elongated objects), and thus its state can be updated by computing at each simulation step the typical second-order ordinary equation that relates position, velocities and accelerations of all nodes with their mass values, spring and damping constants, as well as constraint and external forces acting on the nodes. Hard constraints are given by the collision response when a contact occurs, whereas soft constraints -including goal seeking and path following, among others- are simulated by using penalty forces. An initial estimation of the path is computed by combining an approximate medial-axis transform computation of the workspace (the blood vessels) and a probabilistic roadmap planner. As the method isn't exact, an efficient collision detection module is needed. Due to the high number of possible collisions with the boundary primitives, bounding boxes hierarchies are discarded (also because of the cost of frequently updating the hierarchy as the shape of the robot changes). Instead, an algorithm is presented that exploits both 2.5D overlap tests using the rasterization capabilities of graphics processing units, and the concept of potentially colliding sets.

3.3 Planar objects

Planar objects have two privileged dimensions, whereas thickness, the third one, is negligible for planning purposes. Thin metal plates, cloth, fabric, or paper are some examples of this kind of objects, which exhibit a quite different behavior in each case.

3.3.1 Generic planar objects

General approaches to manipulation of planar flexible objects, like the work in [71], [88], [66], [89] could in fact extend to 3-dimensional objects as well. Random sampling based planning methods are the natural choice for this kind of problems, due to the high dimensionality of the configuration space. A simultaneous computation of rigid transformations and deformations of a free flying flexible surface patch, while avoiding obstacles, to attain a target configuration, is aimed at in [71] within the probabilistic roadmap framework as a first step towards planning its necessary manipulation. The flexible patch is modelled as a low-degree Bézier surface, which is defined by a reduced set of control points, and 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. Samples whose energy values exceed predefined limits are discarded. In [88, 89] two important enhancements are introduced: grasping constraints are explicitly captured by constraining the position and tangent of given sets of points (two boundary edges, for instance), and a more realistic elastic deformation model, which takes the material properties directly into account (Young modulus and Poisson ratio). Together with the grasping conditions, elastic energy minimization determines the position of the other control points, and consequently the deformation of the plate. A different enhancement is introduced in [66]: the medial axis of the workspace is used for guiding the sampling of the probabilistic roadmap method. This medial axis (actually an approximation) is computed as a curve in the plane: the walls of the workspace, and the flexible surface itself, are considered to be perpendicular to this plane (an extension to full 3D could be contemplated, using medial axis computations like [124, 120]). The advantage of biasing sampling towards the medial axis is to avoid the filling of large empty regions of the workspace with milestones, while also placing more samples in tight areas. Configurations of the object on or near the medial axis are obtained by fitting the manipulation constraints on it.

3.3.2 Cloth simulation and handling

There is a category of planar objects that exhibit a quite differentiated and complex behavior; they can be grouped under the generic denomination of cloth. The few robotic works that have dealt with fabric pieces follow sensor-based approaches, and will thus be reviewed in a companion paper. No model-based cloth-specific planning has been undertaken in the robotics community. The closest work has to be found in Computer Graphics, where large efforts have been devoted to cloth simulation, since a couple of decades ago. Cloth simulation appears in items like flaming flags, vibrating sails, undulating curtains (see, for example, [140] for a waving flag as a particular instance of a generic model for elastic deformable objects), but most of the work has been devoted to the drape of cloth over solid objects [37, 31, 21, 73, 35, 158] and to the interaction between pieces of cloth and animated characters wearing them [33, 21, 35, 75, 146, 104]. Some of the challenges faced by these works are very garment-specific, mainly those related to the involved interaction with this other highly deformable entity which is the human body. But others are common to the handling of any piece of fabric and thus concern also a potential robotic manipulation. 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 [104]. Although the CG community is more concerned about the *appearance* of realism than about reproducing exactly the real behavior, and purely geometrical fakes of cloth behavior have been developed since the catenaries-based approach in [155], physically based techniques are increasingly popular. They provide the developed modelling and simulation algorithms with a high degree of reliability. This, together with the obvious search for computational efficiency, render these techniques suitable to robotic planning applications. Standard protocols, as the Kawabata Evaluation System (KES) [82, 83], 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, which may be quite different from one kind of fabric to another (a trained eye may distinguish the type of fabric from the way it drapes [31], see also [73] for a work relating the fabric drape coefficient from the Cusick drapemeter with the mechanical properties from KES tests).

Fabric has been modelled mainly with continuum models, implemented with finite difference [140] and finite elements methods [37, 55, 21, 139], and with discrete models like particle systems [31, 54] and mass-spring models (where also cross-springs connecting non-immediate neighbors are necessary for modelling the flexural resistance of cloth) [35, 146, 118, 104], see Figure 14.

Continuum models face serious drawbacks when applied to cloth, due to the high computational requirements of these methods (very fine meshing to produce large deformations) and difficult integration of highly variable constraints, as pointed out in [104], mainly the high instability and non-linearity of the buckling behavior (formation of wrinkles) [35]. 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 [104] (see this reference for recommendations about the best suited

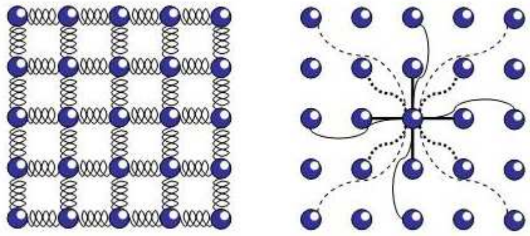

Figure 14: A purely rectangular pattern as in the mass-spring system on the left is not sufficient for simulating the complex behavior of cloth. Diagonal springs, as well as springs connecting non-immediate neighbors, as shown on the right, have to be added (such patterns are used in [35], whereas in [118] diagonal springs are restricted to immediate diagonal neighbors). In [118] the springs drawn in bold lines are called “structural springs”, those in bold dots “shear springs”, and those in plain fine lines “flexural springs”.

numerical methods for different applications). Viscosity (other than the *material intrinsic damping* of cloth [35]) is often added to the model to speed up convergence and to enhance numerical stability [35, 104]. Most methods relate cloth deformation to an energy function, although attempts at incorporating measurable mechanical properties directly into the model also exist [158].

As pointed out in [146], constraints influencing the motion of cloth are of two types: continuous constraints like the material properties of fabric (internal constraints) and forces like gravity and wind (external constraints), and discontinuous constraints that arise from collisions with other objects. 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.

Collision detection and response is one of the most crucial and also time-consuming modules of a cloth simulation system. The number of contact points with other objects in the environment can be very large, extremely high in the case of garments and the underlying moving bodies. Furthermore, folding, draping, or buckling behaviors give rise to an equally large number of self-collisions. This means that two types of collisions have to be handled: surface-volume and surface-surface collisions. The latter also arise when two different pieces of fabric overlie. Whereas in the first type of collisions the volumetric nature of one of the colliding objects imposes a clear inside-outside orientation on its boundary, and thus it’s straightforward to determine when the cloth is penetrating the underlying object, the same does not hold for two surfaces, as shown in Figure 15.

Surface-volume collisions can be efficiently detected using hierarchies of bounding volumes, like Axis-Aligned Bounding Boxes (AABB) hierarchies in [142], [15] or hierarchies of k-dops in [146, 143, 144, 110]. A recent development addressed specifically at deformable bodies is the dynamic bounding volume hierarchy described in [93]. Aiming at larger simulation timesteps, thus needing some prognostics about imminence of collision, proximity is also checked for in [110] by an oriented k-dop “inflation” (where the orientation is provided by the velocity vector). As for surface-surface collisions, the same bounding-

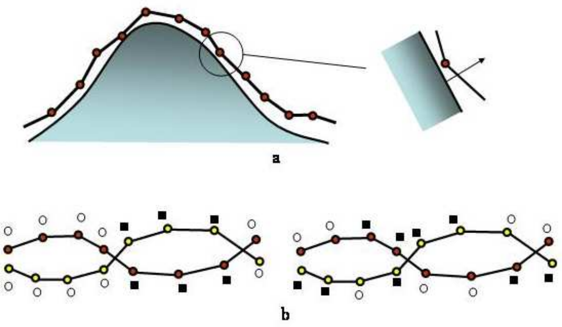

Figure 15: (a) Penetration of surface nodes in the underlying object can be easily detected in surface-volume collisions. (b) In surface-surface collisions there is no “interior”. The small circles label nodes that are “on the right side”, black square those “on the wrong side” of each other surface. Consistent collision orientations are depicted on the left, unconsistent on the right (from [143]))

volume hierarchy framework has been used in [142, 146, 110] together with a curvature (and contour intersection) test, see Figure 16.

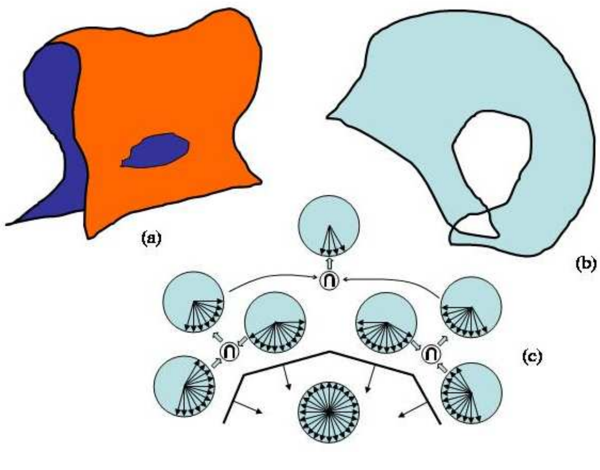

Figure 16: Self-collisions may only appear if (a) the surface is curved enough, or (b) if it is not, the contour of the piece projected on the plane given by the common normal of the surface does intersect. This common normal direction can be found by logically AND-ing the bitfields of sampling directions along the representing hierarchy (c) (from [142]).)

Consistency of colliding surfaces as depicted in Figure 15(b) is provided in [146, 143] by grouping together the collisions that belong to the same “contact region” and computing statistically the most probable orientation, which is assigned to all collisions inside the same region. This is combined with remnant algorithms that keep track of the evolution of collisions along time. However, history-dependent strategies may hinder the recovery from wrong configurations if they are erroneously remembered to be correct, as pointed out in [22]. An alternative strategy of identifying closed intersection contours of colliding regions is described there, these regions are reconstructed with a flood-filling algorithm and brought back to a non-colliding situation. However, it is not always possible to have unambiguously defined intersection contours, and when the surface boundaries intersect, it makes no sense to talk about the orientation of the intersecting surfaces. These limitations are overcome in [145] with a strategy that just tries to minimize the length of the intersection contours.

Collision response is the mechanism by which intersection situations are resolved. This can be implemented as a direct correction of the state of the system, i.e., position (at a minimum collision distance) and velocities (so as to avoid coming closer) of the colliding particles, and a force (or acceleration) correction that enforces these kinematical constraints [146, 143], or computing a particle acceleration correction (using weighted sums of positions, velocities and accelerations of the whole set of colliding particles) which translates to a force correction factor (the collision force) to be added to the mechanics of each particle [144].

As for other related areas, see also [69] for CAD tools for garment design, this survey includes a table summarizing different approaches to fabric modelling, ranging from 1978 to 1996, from Textile engineering, Computer Graphics and CAD.

3.3.3 Origami (paper folding), carton folding

Paper is a stuff whose manipulation mastering by a robot, like in the case of cloth, could open the road to a large number of applications. Unlike the case of fabrics, however, where folds are highly unstable, a fold (or *crease*) in a sheet of paper 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 [45], and, in particular, may touch themselves (flat foldings in flat origami). The *crease pattern* refers to the arrangement of creases on the flat paper sheet, indicating whether each particular crease folds as a *mountain* or as a *valley*. Particular categories of folds are *simple folds* and *book folds*, which belong to the basic skills of *Pureland Origami* (a concept coined by John Smith in the 1970's). *Book folds* reflect a set of facets of one side of the crease (or set of collinear creases) onto the other, that is, they are ± 180 deg folds. *Simple folds* (symmetrical folds that leave the same amount of paper on both sides of the crease) are a particular case of book 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. Figures 17 and 18 illustrate these concepts.

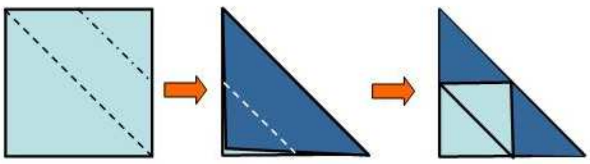

Figure 17: Crease pattern and the corresponding folds. The standard convention of noting valleys by dashed and mountains by dash-dot lines is followed. A simple fold is followed by a book fold.

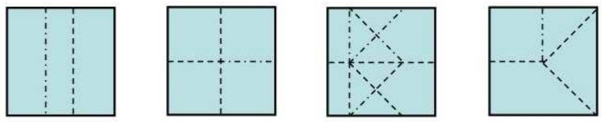

Figure 18: Crease patterns corresponding to flat-foldable examples. The first three, from left to right: foldable by one-layer but not by all-layers folds, foldable by all-layers but not by one-layers folds, foldable by some-layers but neither by one- nor by all-layers folds. These three examples are flat foldable by book-folds, whereas the fourth one can only be folded by general origami (non Pureland) folds.

The generic problem of folding and unfolding has received recently increased attention in the computational geometry community. Surveys have been published focusing on linkages, origami and polyhedra [45, 46]. Linkages are rigid bars connected by rotational joints and the *fundamental questions* refer to whether any simple configuration can be folded into a canonical one: whether any open polygonal chain in any configuration can be completely straightened, any closed polygonal chain convexified, and any polygonal tree acquire a flat configuration. In the plane, the answer is affirmative for the first two and negative for the third (due to the existence of "locked trees"). In 3D, all three fundamental questions have to be answered with a "no" (consider, for example, "almost knots" that cannot be untangled), whereas the answer is always affirmative for 4D or more (see [45] for the specific references). These results (for 2D and 3D) and the algorithms devised to perform such configuration changes (which can be generalized) whenever possible are of great interest not only for planning the manipulation of such kind of objects but also for robot motion planning (where joint limits constitute additional constraints). 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 (Figure 19). In [13] 1D foldability is analyzed before proceeding to map folding, a special case of origami addressed below. Polyhedra folding, on the other hand, concerns the well-known construction method of polyhedra by cutting out a so-called flat *net* or *unfolding*, fold it up and glue the boundary edges together. The reverse is also object of research, namely to compute a flat unfolding for a given polyhedron. These problems are closely related to linkages and origami, see [45, 46] for more details.

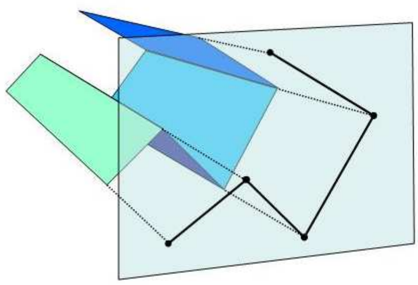

Figure 19: Relationship between origami and simple planar polygonal chains.

As for paper folding, a categorization of the young field of *computational origami* divides results into *universality results* (of the type "any polygonal silhouette can be folded out of a sufficiently large piece of (rectangular) paper"), *efficient* (polynomial time) *decision algorithms* to determine whether a given object is foldable (for example, to decide whether a map can be folded by a sequence of simple folds), and *computational intractability results* [45, 46]. The *origami folding* problem, i.e., to determine whether a particular crease pattern on the unfolded paper sheet can be folded into anything, is usually intractable, whereas the *origami design* problem -that is, to fold a sheet of paper into an object with some target specifications, mostly concerning shape- can be solved in general (although most algorithms up to date do not lead to practical foldings) [45]. A practical method for origami design is the *tree method*, whose algorithmic formulation and corresponding implementation, TreeMaker, is due to Robert Lang [92] (see also [90, 91]). Origami design does also address the fold-and-cut problem (obtaining a given shape by folding, making one straight cut, and unfolding the paper), and is also surveyed in [45, 46].

Returning to the origami *folding* problem, 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 [28], global foldability -computing the overlap order of the crease faces that fold to a common portion of the plane, in words of [45]- is NP-hard in general. A special (simpler) case is that of map folding: to determine whether a flat folding exists for a given crease pattern of mountains and valleys aligned with the sides of the rectangular sheet to be folded, via a sequence of simple folds (one-, some- or all-layers). It is shown in [13] that the decidability problem solves in deterministic linear time for the some-layers case and $O(n \log n)$ for the all-layers case (a randomized linear algorithm is also provided). See also [19] for some results concerning the foldability of paper bags.

A robotic origami folding system is described in [18], [20]. 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 [74, 18, 20] for necessary conditions for flat-foldability. The machine consists of a 4-dof SCARA robot equipped with a vacuum pad for positioning the sheet of paper and a blade, which presses the paper into a clamp that folds the crease. This construction conditions the type of allowed folds to be book folds. A planner determines the sequence of book folds given the crease pattern on the sheet and the desired stacking of facets. The state change experimented by the object (the modification of its shape) does not occur while the robot is holding the object, the robotic arm just performs pick-and-place (or machine feeding) operations. However, considering the system as a whole, planning the motions of the mobile parts has to take these deformations into account, mainly for avoiding collisions. Furthermore, the sequence of transit and transfer motions is obviously determined by the order in which these shape modifications have to be performed, but it is also true that if there are different alternative ways to come to the same goal state, optimality criteria concerning the number and type of motions may guide the choice of the most convenient alternative.

Wrapping paper or similar stuff around volumetric objects has received little attention from the robotics community, despite -or maybe because- it constitutes a challenge involving advanced manipulation skills (not to confound with some human-assisted devices in industry known as "robotic wrapping systems" or "wrapping robots" for stretch wrapping plastic foils around objects). Computational geometry scientists and mathematicians are concerned about universality results with a strong theoretical flavor, like whether a given shape can be wrapped with a sufficiently large rectangle of paper ([44]), (see also [34, 30, 45, 46, 43] for similar problems). Some of these works provide also the folding patterns, and could be inspiring for deriving folding strategies that maybe are suboptimal but possibly more efficient from the manipulation standpoint.

Packing in boxes is more familiar to roboticists, as it is one of the typical pick-and-place operations performed by robots in industry. The subject of constructing the box itself, however, is quite unusual in robotics. Carton folding is related to origami, but even more to folding linkages. This has driven the work in [102] 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 valid paths of such polyarticulated branching manipulators, i.e., the valid folding sequences for the carton. The only considered collisions (to be avoided by the planner) are self-collisions (between the carton panels). The selection of the final sequence is done interactively with the designer of the fixtures layout (interchangeable -to allow quick changes- fixtures, that cause the carton panels to rotate, are used), where the motions of the robot that manipulates the carton are taken into account. The authors point at the potential use of their techniques to sheet metal bending and to the design of 3D MEMS structures from 2D hinged elements.

Nanostructure fabrication processes, like lithography -the most extended one-, is typically bidimensional. The conquest of the third dimension (see Figure 20) enables the construction of new devices with new properties, and an efficient way to do this is *Nanostructure Origami* [137]. A 2D substrate is patterned, creating creases on it and subsequently folding the hinges to spatial configurations. 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. To this end, an energy minimization method is applied in [137]. The dynamics of an *accordion origami* constructed (i.e., folded) by applying the *stress actuation method* on a patterned membrane is also studied in this reference.

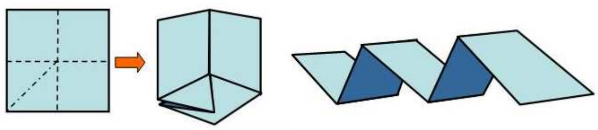

Figure 20: 3D structures obtained by folding. On the left, a corner-like structure. On the right, accordion origami.

3.3.4 Sheet metal bending

Another closely related area is sheet metal bending. Like in the preceding applications, the shape of the manipulated objects changes during the process. Nonetheless, there are also important differences: bending angles remain fixed once bent, simultaneous bends are in most cases collinear [102, 68], and the sheet metal remains piecewise planar during the bending process [45]. 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 21).

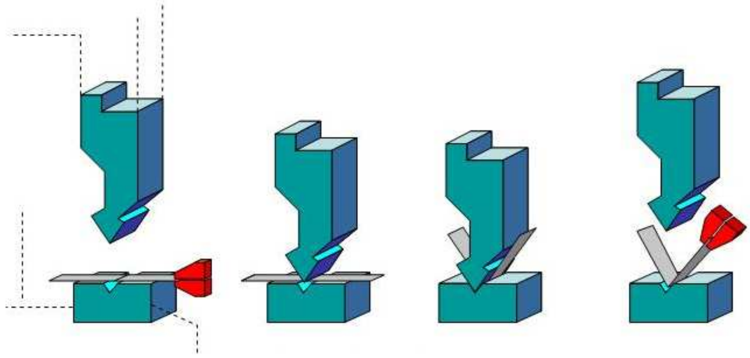

Figure 21: Bending operations, as described in the text, from left to right. During the first and last operations, manipulation of the part has to take possible collisions with the press into account (where the shape of the part is different in the two cases).

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 22 and 23).

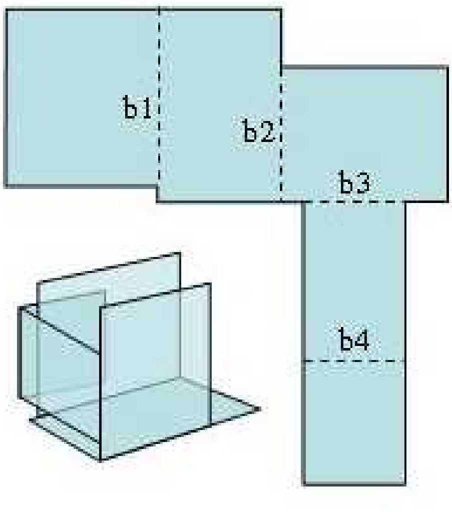

Figure 22: Sheet metal blank with bend lines, and final product. Collision constraints impose partial orderings on the bending sequence: for example, b1 cannot be bent after b3 and b4.

Ordering constraints arise from geometrical limitations (to avoid collisions between the part, the machine and the robot, and also self-collisions of the part while bending),

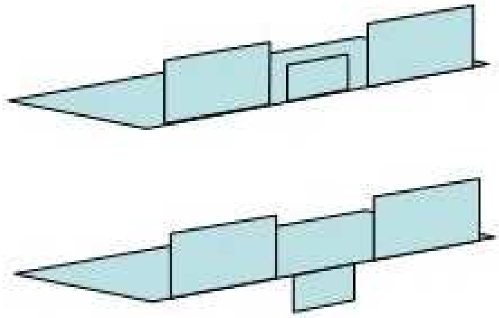

Figure 23: With an adequate arrangement of punches and dies, all three bendings can be performed in a single step for the part depicted above, whereas for the one shown on the bottom requires an additional reorientation of the part.

from tolerance specifications, as well as from optimality criteria. Collisions involving the displacement of the robot with or without the part may be determined with standard collision detection tools [100, 77], but the collisions that may arise due to the change of the shape of the part while bending require a specific treatment (Figure 24).

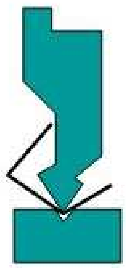

Figure 24: During bending, a collision occurs between the part and the punch.

This issue is surveyed in [52]. There are contributions based on the computation of simplified sweep volume collision verification from the volume swept out by the sheet flanges (as if they were rotating around the bend line) [56] or from equivalent fictitious die and punch rotations [42]. Another alternative is the approach followed in [49] where large amounts of potential collision tests are pruned away by considering the sheet flanges distributed in groups divided by the bend line (intra-group collisions cannot take place). Constraints on admissible tolerances may reject specific sequences of bending operations due to the accumulation of errors. See again [52] for a brief survey of works addressing tolerance and bending sequences. As for soft optimality criteria, they are mostly concerned with minimizing the manipulation time and efforts. Some work exists were ergonomic considerations are reflected in heuristic cost and penalty functions, and which could also apply to the case where the piece is manipulated by the robot: number of workpiece orientation changes [76], combined rotations [41], fixed penalties for each rotation direction [131], or combined penalties for stability and manipulability attributes [119]. Manipulability by robots has also been addressed specifically in other works: in [12] the best grasping positions and repositioning timings are given in advance and act as constraints for bending sequencing. In [153, 68] a specific planner is in charge of the motion of the robot, which requires a specific treatment of fine motion planning of the transfer motions next to the tool, with the particularity that the displaced piece has a different shape before and after the bending.

In sum, sequencing of bending operations is a combinatorial problem, subject to hard constraints and optimality criteria. 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. Explicit rules are of the type "workpiece rotations along two axes must be avoided between consecutive bends", and are aimed at minimizing the material handling time together with a systematic tolerance verification [40, 41] (tool selection is tuned to avoid part-tool collisions if detected in a sequence). As for the implicit encoding by precedence constraints, this can be done in the form of a language like in [68] (including wildcards, for example "b1 precedes b2 with any number of bends before b1 or after b2 and exactly one before b2" is written (*b1?b2*)), or in the links of a directed graph whose nodes are the individual bends [60]. Alternatively, the sequence can be computed by a search in a state graph as in [48, 51] together with reduction techniques like redundancy elimination and pre-processing of the part geometry (i.e., considering possible constraints). It is formulated as a *Travelling Salesman Problem* in [50] and solved with a depth-first branch-and-bound strategy. The standard heuristic A* algorithm is applied to search in a state-tree, which displays a "folding-by-unfolding" approach [131]. Costs express criteria like the required number of tool changes, stability of the placed part, etc. and rules are provided for the estimated cost. In [119] this constitutes the initial solution in a two-stage branch-and-bound search algorithm (further optimization is based on costs-to-come only). A* is also used in [153, 68] in a distributed architecture where modules specialized in tool selection, grasping and motion planning (as mentioned above) provide constraints to the central planner, evaluate partial solutions and check the final sequence for feasibility. Constraint programming arises also as a natural computation paradigm, and is used in [105, 141, 106, 85]. For a detailed survey on the topic, see [52].

The linear counterpart of sheet metal bending is wire or tube bending. It has been found to be closely related to linkage folding/unfolding, and [14] provide an efficient algorithm do determine if a "linkage" can be straightened including the restrictions that each "joint" can be altered at most once, and folding must be done sequentially from one or both ends of the linkage.

4 Conclusions

Planning the manipulation activity of a robot goes far beyond of just planning its motions. The manipulated object is not just an occasional extension of the robot's geometry, to be taken into account to avoid collisions with the environment, but it has its own behavior which has to be considered in the whole planning process. This becomes evident even in the first works dealing with the manipulation of rigid objects, where planning occurs in a *composite* configuration space, i.e., in a combination of the configuration spaces of the robot *and* the manipulated object. In these first attempts, stemming from the Computational Geometry and the Robot Motion Planning contexts, transit and transfer motions have to be planned in two subsets of the composite configuration space, namely $\mathcal{C}_{placement}$ and $\mathcal{C}_{grasp}$, linking the connected components shared by these two subsets. Clearly, the behavior of the manipulated object is captured in these spaces. In particular, $\mathcal{C}_{placement}$, the set of allowable placements of the object, one of whose most obvious instances is $\mathcal{C}_{stable}$, the set of configurations where the object is stable under gravity. Planning methods are the same as in Robot Motion Planning: exact cell decompositions and probabilistic sampling methods, and efficient algorithms for solving closed kinematic chains have demonstrated to be of special relevance.

The classical formulation of manipulation planning still leaves space for uncommon robotic manipulations. For example, carrying an open recipient with liquid can be conceived as manipulating a rigid object (the glass) where $\mathcal{C}_{placement}$ equals $\mathcal{C}_{stable}$ for the glass with an additional constraint forbidding configurations that result in pouring out the liquid (and also $\mathcal{C}_{grasp}$ will be constructed so as to avoid inclining the glass too much). Nonetheless, if the task includes transferring the liquid to other recipients, geometry will not suffice: the rheological behavior of the fluid will have to be described by a model that can be used by the planning algorithms.

This happens with a whole class of solids: deformable objects display a complex behavior when they are manipulated, and the efficient inclusion of this behavior in the planning equation constitutes a research challenge. This survey has tried to capture the wealth and complexity of planning the manipulation of deformable objects, by not only describing the algorithms that from a motion planning perspective have been developed for general models of this kind of objects, but also by focusing on a series of specific objects and applications, with predominant linear and planar geometry. While such simpler geometry admits a more tractable modelling of the objects to manipulate, the specific nature of the intended applications introduces aspects to be considered during planning that transcend mere pick-and-place. At the same time, these specifications may help to reduce the search space for manipulation by guiding the process towards feasible solutions.

The choice of linear and planar deformable types of objects dealt with in this survey may seem somehow arbitrary. However, this selection has not only been conditioned by the amount of available references found in literature, but also because they correspond to quite representative applications in Robotics, like assembly, or because these objects stand for a whole category, when considering their deformation behavior. Ropes and cloth, for example, are the linear and planar counterparts of flexible objects (as defined at the beginning of Section 3), and steered needles and the general plates dealt with in Section 3.3.1 can both be considered pure elastic bodies. Furthermore, their inclusion in the same survey may be cross-fertilizing: for example, the relevance of topology in knotting ropes may be inspiring for applications concerning the handling of cloth.

The issue of manipulation planning has been treated in this survey assuming that a model of the object suitable for planning was already available (in fact, the section devoted to cloth, for instance, concentrated mainly on its modelling, as little has been done in planning), and its behavior was predictable. The type of plans generated prescribe a sequence of motions and actions at a level of certain abstraction. For example, in knotting/unknotting manipulation, the sequence of crossing/uncrossing operations, together with the grasping points, the moving directions, and the directions of approach constitute what in [152, 149] is called a *qualitative manipulation plan*. An effective implementation in a real setting requires, of course, sensory feedback on the attainment of the different subgoals by the manipulating robot: a visual system that checks whether a specific pulling of a rope segment by the robot has effectively led to the intended crossing state. The same happens with the other applications and types of objects as well. For instance, robotic manipulation of cloth has been dealt with mainly on a sensory feedback basis in literature. Also the work of [122] mentioned in Section 3.2.1 relies on vision and force feedback to effectively verify the completion of contact transitions in a real-world implementation.

Another assumption was that the purposes of the manipulation and the main layout of the manipulation tasks were already established. Such an assumption may compromise the autonomy of the robot, specially if it has to carry out its manipulation tasks in a non-controlled environment. A higher level of abstraction, as well as reasoning and learning capabilities come into consideration. In Section 2.3 the subject of how to link together the different levels of abstraction, or, in other words, how to connect manipulation planning as reviewed in this survey to task-level planning has been briefly treated. Learning to manipulate, on the other hand, is a very active subject of research, and relies heavily on an adequate processing of sensorial information. Thus, sensor-based robotic manipulation, comprising not only learning but also fixed sensor-feedback control algorithms, deserves another survey so as to provide a complete vision on the subject.

Acknowledgments

This work is supported by the EU PACO-PLUS project FP6-2004-IST-4-27657 and by the Catalan Research Comission through the Consolidated Robotics Group.

References

1. Abaci, F., Mortara, M., Patane, G., Spagnulo, M., Vexo, F., and Thalmann, D. Bridging geometry and semantics for object manipulation and grasping. In *Proceedings of Workshop towards Semantic Virtual Environments (SVE 2005)* (march 2005), pp. 110–119.

2. Abegg, F., Engel, D., and Wörn, H. Manipulating deformable linear objects - vision-based recognition of contact state transitions. In *The Ninth International Conference on Advanced Robotics (ICAR99)* (october 1999), pp. 135–140.

3. Abegg, F., Engel, D., and Wörn, H. A robust algorithm for segmenting deformable linear objects from video image sequences. In *International Conference on Pattern Recognition (ICPR 2000)* (september 2000), vol. 4, pp. 4756–4759.

4. Acker, J., and Henrich, D. Manipulation of deformable linear objects: From geometric model towards program generation. In *Proceedings IEEE International Conference on Robotics and Automation, ICRA 2005* (2005), pp. 1541–1547.

5. Adams, C. C. *The Knot Book: An Elementary Introduction to the mathematical Theory of Knots*. W. H. Freeman and Co., 1994.

6. Ahuactzin, J., Gupta, K., and Mazer, E. Manipulation planning for redundant robots: A practical approach. *International Journal of Robotics Research 17*, 7 (july 1998), 731–747.

7. Alami, R., Laumond, J. P., and Siméon, T. Two manipulation planning algorithms. In *WAFR: Proceedings of the workshop on Algorithmic foundations of robotics* (Natick, MA, USA, 1995), A. K. Peters, Ltd., pp. 109–125.

8. Alami, R., Siméon, T., and Laumond, J.-P. A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps. In *The fifth international symposium on Robotics research* (Cambridge, MA, USA, 1990), MIT Press, pp. 453–463.

9. Alterovitz, R., Goldberg, K., and Okamura, A. M. Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2005* (2005), pp. 1652–1657.

10. Alterovitz, R., Lim, A., Goldberg, K., Chirikjian, G. S., and Okamura, A. M. Steering flexible needles under markov motion uncertainty. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005* (2005), pp. 120–125.

11. Anshelevich, E., Owens, S., Lamiraux, F., and Kavraki, L. E. Deformable volumes in path planning applications. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000* (2000), pp. 2290–2295.

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

13. Arkin, E. M., Bender, M. A., Demaine, E. D., Demaine, M. L., Mitchell, J. S. B., Sethia, S., and Skiena, S. S. When can you fold a map? *Comput. Geom. Theory Appl. 29*, 1 (2004), 23–46.

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

15. Baciu, G., and Wong, W. S.-K. Image-based collision detection for deformable cloth models. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (2004), 649–663.

16. Badler, N., Bindiganavale, R., Allbeck, J., Schuler, W., Zhao, L., Lee, S.-J., Shin, H., and Palmer, M. Parameterized action representation and natural language instructions for dynamic behavior modification of embodied agents. In *AAAI Spring Symposium* (2000).

17. Badler, N., Bindiganavale, R., Bourne, J., Palmer, M., Shi, J., and Schuler, W. A parameterized action representation for virtual human agents. In *Workshop on Embodied Conversational Characters* (october 1998).

18. Balkcom, D. J. *Robotic origami folding.* PhD thesis, Carnegie Mellon University, august 2004. RI TR 04-43.

19. Balkcom, D. J., Demaine, E. D., and Demaine, M. L. Folding paper shopping bags. In *Abstracts from the 14th Annual Fall Workshop on Computational Geometry* (Cambridge, Massachusetts, November 19–20 2004), pp. 14–15.

20. Balkcom, D. J., and Mason, M. T. Introducing robotic origami folding. In *Proceedings of the IEEE International Conference on Robotics and Automation* (april 2004), pp. 3245–3250.

21. Baraff, D., and Witkin, A. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 43–54.

22. Baraff, D., Witkin, A., and Kass, M. Untangling cloth. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 862–870.

23. Barraquand, J., and Ferbach, P. Path planning through variational dynamic programming. In *Proc. 1994 IEEE Int. Conf. Robotics and Automation* (1994), pp. 1839–1846.

24. Barraquand, J., and Ferbach, P. A penalty function method for constrained motion planning. In *Proc. 1994 IEEE Int. Conf. Robotics and Automation* (1994), pp. 1235–1242.

25. Barraquand, J., and Latombe, J.-C. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research 10*, 6 (Dec. 1991), 628–649.

26. Bayazit, O. B., Lien, J.-M., and Amato, N. M. Probabilistic roadmap motion planning for deformable objects. In *Proc. IEEE Int. Conf. on Robotics and Automation* (2002), IEEE, pp. 2126–2135.

27. Benosman, M., Vey, G., Lanari, L., and Luca, A. D. Rest-to-rest motion for planar multi link .exible manipulator through backward recursion. *ASME Journal of Dynamic Systems Measurement and Control 126* (2004), 115–123.

28. Bern, M., and Hayes, B. The complexity of flat origami. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 1996), Society for Industrial and Applied Mathematics, pp. 175–183.

29. Bessiere, P., Ahuactzin, J.-M., Talbi, E.-G., and Mazer, E. The ariadne's clew algorithm: global planning with local methods. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)* (july 1993), vol. 2, pp. 1373–1380.

30. Biedl, T. C., Demaine, E. D., Demaine, M. L., Lubiw, A., and Toussaint, G. T. Hiding disks in folded polygons. In *Proceedings of the 10th Canadian Conference on Computational Geometry (CCCG'98)* (Montréal, Québec, Canada, August 10–12 1998).

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

32. Brown, J., Latombe, J.-C., and Montgomery, K. Real-time knot-tying simulation. *Vis. Comput. 20*, 2 (2004), 165–179.

33. Carignan, M., Yang, Y., Thalmann, N. M., and Thalmann, D. Dressing animated synthetic actors with complex deformable clothes. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), ACM Press, pp. 99–104.

34. Catalano-Johnson, M. L., and Loeb, D. Problem 10716: A cubical gift. *American Mathematical Monthly 108*, 1 (January 2001), 81–82.

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

36. Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, June 2005.

37. Collier, J., Collier, B., OToole, G., and Sargand, S. Drape prediction by means of finite elements analysis. *Journal of the Textile Institute 82*, 1 (1991), 96–107.

38. Cortés, J., Siméon, T., and Laumond, J. A random loop generator for planning the motions of closed kinematic chains using prm methods. In *Proc. 2002 IEEE Int. Conf. Robotics and Automation* (2002), vol. 2, pp. 2141–2146.

39. Şahin, E., Çakmak, M., Doğar, M. R., Uğur, E., and Üçoluk, G. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior 15*, 4 (2007), 447–472.

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

41. de Vin, L., Streppel, A., and Kals, H. Set-up sequencing procedures for sheet bending. In *Proceedings of the 3rd International Conference on Sheet Metal* (Birmingham, UK, june 1995), pp. 86–95.

42. Decubber, C., and Schreurs, P. Werkvoorbereiding voor het plooien van plaatvormige werkstukken. Master's thesis, Department Werktuigkunde, FTW, K.U. Leuven, Belgium, 1991.

43. Demaine, E. D., Demaine, M. L., Iacono, J., and Langerman, S. Wrapping the Mozartkugel. In *Abstracts from the 20th European Workshop on Computational Geometry (EuroCG 2007)* (Graz, Austria, March 19–21 2007), pp. 14–17.

44. Demaine, E. D., Demaine, M. L., and Mitchell, J. S. B. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Computational Geometry: Theory and Applications 16*, 1 (2000), 3–21. Special issue of selected papers from the 3rd CGC Workshop on Computational Geometry, 1998.

45. Demaine, E. D., and O'Rourke, J. A survey of folding and unfolding in computational geometry. In *Combinatorial and Computational Geometry*, J. E. Goodman, J. Pach, and E. Welzl, Eds., vol. 52 of *Mathematical Sciences Research Institute Publications*. Cambridge University Press, August 2005, pp. 167–211.

46. Demaine, E. D., and O'Rourke, J. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, July 2007.

47. Douville, B., Levison, L., and Badler, N. I. Task-level object grasping for simulated agents. *Presence 5*, 4 (1996), 416–430.

48. Duflou, J. Design verification for bent sheet metal parts: a graphs approach. *International Transactions in Operational Research 4*, 1 (1997), 67–73.

49. Duflou, J., Dung, H., and Kruth, J.-P. A collision detection algorithm for the verification of sheet metal bending set-ups. In *Proceedings of the 6th International Conference on Sheet Metal* (Enschede, Netherlands, april 1998), vol. 2, pp. 209–220.

50. Duflou, J., Kruth, J.-P., and Oudheusden, D. V. Algorithms for the design verification and automatic process planning for bent sheet metal parts. In *Annals of the CIRP* (1999), vol. 48/1, pp. 405–408.

51. Duflou, J., Oudheusden, D. V., Kruth, J.-P., and Cattrysse, D. Methods for the sequencing of sheet metal bending operations. *International Journal of Production Research 37*, 14 (september 1999), 3185–3202.

52. Duflou, J. R., Váncza, J., and Aerens, R. Computer aided process planning for sheet metal bending: a state of the art. *Comput. Ind. 56*, 7 (2005), 747–771.

53. Dwivedya, S. K., and Eberhard, P. Dynamic analysis of flexible manipulators, a literature review. *Mechanism and Machine Theory 41*, 7 (july 2006), 749–777.

54. Eberhardt, B., Weber, A., and Strasser, W. A fast, flexible, particle-system model for cloth draping. *IEEE Comput. Graph. Appl. 16*, 5 (1996), 52–59.

55. Eischen, J. W., Deng, S., and Clapp, T. G. Finite-element modeling and control of flexible fabric parts. *IEEE Comput. Graph. Appl. 16*, 5 (1996), 71–80.

56. Franke, V. Automation of tool planning for bent components. In *Proceedings of the 3rd International Conference on Sheet Metal* (Birmingham, UK, april 1995), pp. 35–44.

57. Fukuda, T., Matsuno, T., and Arai, F. Flexible object manipulation by dual manipulator system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00)* (2000), vol. 2, pp. 1955–1960.

58. Gayle, R., Lin, M., and Manocha, D. Constraint based motion planning of deformable robots. In *Proceedings of The IEEE International Conference on Robotics and Automation (ICRA)* (April 2005), IEEE Press, pp. 1046–1053.

59. Gayle, R., Segars, P., Lin, M. C., and Manocha, D. Path planning for deformable robots in complex environments. In *Robotics: Science and Systems 2005* (2005), pp. 225–232.

60. Geiger, M., Hoffmann, M., and Kluge, B. Inferenzmaschine fr ein biegestadien-planungssystem. *Zeitschrift fr Wirtschaftliche Fertigung und Automatisierung 87*, 5 (1992), 261–264.

61. Gibson, J. J. *The Ecological Approach to Visual Perception.* Lawrence Erlbaum Associates, 1979.

62. Gibson, S., and Mirtich, B. A survey of deformable modeling in computer graphics. Tech. Rep. TR-97-19, Mitsubishi Electric Research Lab, Cambridge(MA), November 1997.

63. Gonçalves, L. M. G., Kallmann, M., and Thalmann, D. Programming behaviors with local perception and smart objects: An approach to solve autonomous agents tasks. In *Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'01)* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 184–191.

64. Gravot, F., Alami, R., and Siméon, T. Playing with several roadmaps to solve manipulation problems. In *IEEE/RSJ International Conference on Intelligent Robots and System* (Natick, MA, USA, 2002), vol. 3, pp. 2311–2316.

65. Gravot, F., Alami, R., and Siméon, T. *Robotics Research*, vol. 15/2005 of *Springer Tracts in Advanced Robotics.* Springer Berlin / Heidelberg, 2005, ch. aSyMov:A Planner That Deals with Intricate Symbolic and Geometric Problems, pp. 100–110.

66. Guibas, L. J., Holleman, C., and Kavraki, L. E. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 99* (october 1999), vol. 1, pp. 254–259.

67. Gupta, K. Motion planning for "flexible" shapes (systems with many degrees of freedom): A survey. *The Visual Computer 14*, 5/6 (1998), 288–302.

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

69. Hardaker, C. H. M., and Fozzard, G. J. W. Towards the virtual garment: three-dimensional computer environments for garment design. *International Journal of Clothing Science and Technology 10*, 2 (1998), 114–127.

70. Hergenröther, E., and Dähne, P. Real-time virtual cables based on kinematic simulation. In *Proceedings of the WSCG 2000* (2000).

71. Holleman, C., Kavraki, L. E., and Warren, J. D. Planning paths for a flexible surface patch. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 1998* (1998), pp. 21–26.

72. Hopcroft, J. E., Kearney, J. K., and Krafft, D. B. A case study of flexible object manipulation. *Int. J. Rob. Res. 10*, 1 (1991), 41–50.

73. Hu, J., and Chan, Y.-F. Effect of fabric mechanical properties on drape. *Textile Research Journal 68*, 1 (1998), 57–64.

74. Hull, T. On the mathematics of flat origamis. In *Congressus Numerantium,100* (1994), pp. 215–224.

75. Igarashi, T., and Hughes, J. F. Clothing manipulation. In *15th Annual Symposium on User Interface Software and Technology, ACM UIST'02* (2002 2002), pp. 91–100.

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

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

78. Kabul, I., Gayle, R., and Lin, M. C. Cable route planning in complex environments using constrained sampling. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2007), ACM Press, pp. 395–402.

79. Kallmann, M. *Object Interaction in Real-Time Virtual Environments*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2001.

80. Kallmann, M., and Thalmann, D. Direct 3d interaction with smart objects. In *ACM Symp. on Virtual Reality, Software and Technology* (december 1999), pp. 124–129.

81. Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics & Automation 12*, 4 (June 1996), 566–580.

82. Kawabata, S. *The Standardization and Analysis of Hand Evaluation*, 2nd edition ed. Hand Evaluation and Standardisation Committee (HESC), Textile Machinery Society of Japan, Osaka, Japan, 1980.

83. Kawabata, S., and Niwa, M. Fabric performance in clothing and clothing manufacture. *Journal of the Textile Institute 80*, 1 (1989), 19–50.

84. Koga, Y., and Latombe, J.-C. On multi-arm manipulation planning. In *Proc. 1994 IEEE Int. Conf. Robotics and Automation* (1994), pp. 945–952.

85. Kovács, A., and Váncza, J. Constraint feedback in solving incomplete models: a case study in sheet metal bending. In *Proceedings of STAIRS 2002Frontiers in Artificial Intelligence and Applications* (2002), vol. 78, pp. 109–118.

86. Kuffner, J. J., and LaValle, S. M. 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.

87. Ladd, A., and Kavraki, L. Motion planning for knot untangling. In *Algorithmic Foundations of Robotics V, Springer Tracts in Advanced Robotics* (2004), pp. 7–23.

88. Lamiraux, F., and Kavraki, L. E. Path planning for elastic plates under manipulation constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 1999* (1999), pp. 151–156.

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

90. Lang, R. J. A computational algorithm for origami design. In *SCG '96: Proceedings of the twelfth annual symposium on Computational geometry* (New York, NY, USA, 1996), ACM Press, pp. 98–105.

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

92. Lang, R. J. Treemaker. http://www.langorigami.com/science/treemaker/-treemaker5.php4, 2005. Current version TreeMaker 5.

93. Larsson, T., and Akenine-Möller, T. A dynamic bounding volume hierarchy for generalized collision detection. *Computers & Graphics 30*, 3 (June 2006), 451–460.

94. Latombe, J.-C. *Robot Motion Planning*, vol. SECS 0124. Kluwer, Dordrecht, The Netherlands, 1991.

95. Laumond, J. P., and Alami, R. A geometrical approach to planning manipulation tasks: The case of a circular robot and a movable circular object amidst polygonal obstacles. Tech. Rep. 88314, LAAS, Toulouse, 1988.

96. Laumond, J. P., and Alami, R. A geometrical approach to planning manipulation tasks in robotics. Tech. Rep. 89261, LAAS, Toulouse, 1989.

97. LaValle, S. M. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

98. Levison, L. *Connecting planning and acting via object-specific reasoning*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1996.

99. Li, Q., and Payandeh, S. Manipulation of convex objects via two-agent point-contact push. *The International Journal of Robotics Research 26*, 4 (2007), 377–403.

100. Lin, M. C., and Gottschalk, S. Collision detection between geometric models: a survey. In *IMA Conference on Mathematics of Surfaces* (San Diego (CA), May 1998), vol. 1, pp. 602–608.

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

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

103. Lynch, K. M. Issues in nonprehensile manipulation. In *WAFR '98: Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective* (Natick, MA, USA, 1998), A. K. Peters, Ltd., pp. 237–250.

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

105. Márkus, A., and Váncza, J. Process planning with conditional and conflicting advice. In *Annals of the CIRP* (2001), vol. 50/1, pp. 327–330.

106. Márkus, A., Váncza, J., and Kovács, A. Constraint-based process planning in sheet metal bending. In *Annals of the CIRP* (2002), vol. 51/1, pp. 425–428.

107. Mason, M. T. Progress in nonprehensile manipulation. *The International Journal of Robotics Research 18*, 11 (1999), 1129–1141.

108. Mazer, E., Ahuactzin, J. M., and Bessière, P. The Ariadne's clew algorithm. *Journal of Artificial Intelligence Research 9* (Nov. 1998), 295–316.

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

110. Mezger, J., Kimmerle, S., and Etzmuss, O. Hierarchical Techniques in Collision Detection for Cloth Animation. *Journal of WSCG 11*, 2 (2003), 322–329.

111. Moll, M., and Kavraki, L. Path planning for variable resolution minimal-energy curves of constant length. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2005* (2005), pp. 2143–2147.

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

113. Moll, M., and Kavraki, L. E. Path planning for minimal energy curves of constant length. In *Proceedings of The IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans, LA, April 2004), IEEE Press, pp. 2826–2831.

114. Nielsen, C. L., and Kavraki, L. E. A two level fuzzy prm for manipulation planning. In *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (november 2000), pp. 1716–1722.

115. Park, W., Kim, J. S., Zhou, Y., Cowan, N. J., Okamura, A. M., and Chirikjian, G. S. Diffusion-based motion planning for a nonholonomic flexible needle model. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2005* (2005), pp. 4611–4616.

116. Phillips, J., Ladd, A., and Kavraki, L. Simulated knot tying. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2002* (2002), pp. 841–846.

117. Pond, B., and Sharf, I. Motion planning for flexible manipulators. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (1995), vol. 3, pp. 2103–2108.

118. Provot, X. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface '95* (1995), W. A. Davis and P. Prusinkiewicz, Eds., Canadian Human-Computer Communications Society, pp. 147–154.

119. Radin, B., Shpitalni, M., and Hartman, I. Two-stage algorithm for rapid determination of the bending sequence in sheet metal products. In *Proceedings of the ASME Design Automation Conference* (august 1996), pp. 1–12.

120. Ramanathan, M., and Gurumoorthy, B. A tracing algorithm for constructing medial axis transform of 3d objects bound by free-form surfaces. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005 (SMI' 05)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 228–237.

121. Remde, A., Henrich, D., Karl, S., and Wörn, H. Manipulating deformable linear objects - efficient simulation of the workpiece behavior. In *IASTED Int. Conf. on Robotics and Applications (IASTED RA'99)* (October 1999).

122. Remde, A., Henrich, D., and Wörn, H. Manipulating deformable linear objects - contact state transitions and transition conditions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99)* (October 1999), vol. 3, pp. 1450–1455.

123. Remde, A., Henrich, D., and Wörn, H. Manipulating deformable linear objects - force based detection of contact state transitions, 2000.

124. Romero, F., Ros, L., and Thomas, F. Fast skeletonization of spatially encoded objects. In *Proceedings of the 15th IEEE International Conference on Pattern Recognition (ICPR' 00)* (september 2000), pp. 510–513.

125. Saha, M., and Isto, P. Motion planning for robotic manipulation of deformable linear objects. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2006* (2006), pp. 2478–2484.

126. Sahbani, A., Cortés, J., and Siméon, T. A probabilistic algorithm for manipulation planning under continuous grasps and placements. In *Proceedings of the IEEE/IROS International Conference on Intelligent Robots and Systems* (2002), pp. 1560–1565.

127. Schwartz, J. T., and Sharir, M. On the Piano Movers' Problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics 36* (1983), 345–398.

128. Schwartz, J. T., and Sharir, M. On the Piano Movers' Problem: II. General techniques for computing topological properties of algebraic manifolds. *Communications on Pure and Applied Mathematics 36* (1983), 345–398.

129. Schwartz, J. T., and Sharir, M. On the Piano Movers' Problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *International Journal of Robotics Research 2*, 3 (1983), 97–140.

130. Schwartz, J. T., Sharir, M., and Hopcroft, J. *Planning, Geometry, and Complexity of Robot Motion.* Ablex, Norwood, NJ, 1987.

131. Shpitalni, M., and Saddan, D. Automatic determination of bending sequences in sheet metal products. In *Annals of the CIRP* (1994), vol. 43/1, pp. 23–26.

132. Siméon, T., Cortés, J., Sahbani, A., and Laumond, J. A manipulation planner for pick and place operations under continuous grasps and placements. In *Proceedings IEEE International Conference on Robotics and Automation, ICRA 2002* (2002), vol. 2, pp. 2022–2027.

133. Siméon, T., Cortés, J., Sahbani, A., and Laumond, J. A general manipulation task planner. In *Algorithmic Foundations of Robotics V* (2003), J.-D. Boissonat, K. Goldberg, and S. Hutchinson, Eds., Springer - Verlag, pp. 311–328.

134. Siméon, T., Laumond, J.-P., Cortés, J., and Sahbani, A. Manipulation planning with probabilistic roadmaps. *International Journal of Robotics Research 23*, 7 (July–August 2004), 729–746.

135. Siméon, T., Laumond, J.-P., and Nissoux, C. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics 14*, 6 (2000), 477–493.

136. Steedman, M. Formalizing affordance. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society* (2002), pp. 834–839.

137. Stellman, P., Arora, W., Takahashi, S., Demaine, E. D., and Barbastathis, G. Kinematics and dynamics of nanostructured origami. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition* (Orlando, Florida, November 5–11 2005), pp. 541–548.

138. Stilman, M., Schamburek, J.-U., Kuffner, J., and Asfour, T. Manipulation planning among movable obstacles. In *IEEE International Conference on Robotics and Automation (ICRA07)* (april 2007), pp. 3327–3332.

139. Tan, S. T., Wong, T. N., Zhao, Y. F., and Chen, W. J. A constrained finite element method for modeling cloth deformation. *The Visual Computer 15*, 2 (april 1999), 90–99.

140. Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM Press, pp. 205–214.

141. Váncza, J., and Márkus, A. A constraint engine for manufacturing process planning. In *7th International Conference of Principles and Practice of Constraint Programming* (november-december 2001), vol. LNCS, vol. 2239, Springer, pp. 745–759.

142. Volino, P., and Magnenat-Thalmann, N. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. In *Proceedings of Eurographics '94, Computer Graphics Forum* (1994), vol. 13(3), pp. 155–166.

143. Volino, P., and Magnenat-Thalmann, N. Interactive cloth simulation: Problems and solutions. JWS97-B, Geneva, Switzerland, 1997.

144. Volino, P., and Magnenat-Thalmann, N. Implementing fast cloth simulation with collision response. In *CGI '00: Proceedings of the International Conference on Computer Graphics* (Washington, DC, USA, 2000), IEEE Computer Society, p. 257.

145. Volino, P., and Magnenat-Thalmann, N. Resolving surface collisions through intersection contour minimization. *ACM Trans. Graph. 25*, 3 (2006), 1154–1159.

146. Volino, P., M.Courchesne, and Magnenat-Thalmann, N. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of Siggraph '95* (New York, NY, USA, 1995), ACM Press, pp. 137–144.

147. Vosinakis, S., and Panayiotopoulos, T. A task definition language for virtual agents. *Journal of WSCG 11*, 1 (2003), 512–519.

148. Wada, T., Hirai, S., Mori, H., and Kawamura, S. Robust manipulation of deformable objects using model based technique. In *AMDO '00: Proceedings of the First International Workshop on Articulated Motion and Deformable Objects* (London, UK, 2000), Springer-Verlag, pp. 1–14.

149. Wakamatsu, H., Arai, E., and Hirai, S. Knotting/unknotting manipulation of deformable linear objects. *The International Journal of Robotics Research 25*, 4 (2006), 371–395.

150. Wakamatsu, H., Matsumura, T., Arai, E., and Hirai, S. Dynamic analysis of rod like object deformation towards their dynamic manipulation. In *Proceedings of the IEEE/IROS International Conference on Intelligent Robots and Systems* (1997), pp. 196–201.

151. Wakamatsu, H., Tsumaya, A., Arai, E., and Hirai, S. Planning of one-handed knotting/raveling manipulation of linear objects. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004* (april 2004), vol. 2, pp. 1719–1725.

152. Wakamatsu, H., Tsumaya, A., Arai, E., and Hirai, S. Manipulation planning for knotting/unknotting and tightly tying of deformable linear objects. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004* (april 2005), pp. 2505–2510.

153. Wang, C.-H., and Bourne, D. 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.

154. Webster, R. J., Kim, J. S., Cowan, N. J., Chirikjian, G. S., and Okamura, A. M. Nonholonomic modeling of needle steering. *International Journal of Robotics Research 25*, 5-6 (May-June 2006), 509–525.

155. Weil, J. The synthesis of cloth objects. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1986), ACM Press, pp. 49–54.

156. Wilfong, G. Motion planning in the presence of movable obstacles. In *SCG '88: Proceedings of the fourth annual symposium on Computational geometry* (New York, NY, USA, 1988), ACM Press, pp. 279–288.

157. WOLTER, J., AND KROLL, E. Toward assembly sequence planning with flexible parts. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 1996* (april 1996), pp. 1517–1524.

158. WU, Z., AU, C., AND YUEN, M. Mechanical properties of fabric materials for draping simulation. *International Journal of Clothing Science and Technology 15*, 1 (2003), 56–68.

159. YUE, S., HENRICH, D., XU, W. L., AND TSO, S. K. Point-to-point trajectory planning of flexible redundant robot manipulators using genetic algorithms. *Journal of Robotica 20* (2002), 269–280.

160. ZHAO, H., AND CHEN, D. Optimal motion planning for flexible space robots. In *Proceedings of IEEE International Conference on Robotics and Automation* (April 1996), vol. 1, pp. 393–398.

161. ZHENG, Y., PEI, R., AND CHEN, C. Strategies for automatic assembly of deformable objects. In *Proceedings of IEEE International Conference on Robotics and Automation* (April 1991), vol. 3, pp. 2598–2603.