Planning Grasp Strategies That Exploit Environmental Constraints

Clemens Eppner and Oliver Brock

Abstract—In this work we encourage the use of surfaces, edges, etc. – so-called *environmental constraints* – for grasp planning. We present a method that generates high-level plans given depth measurements of arbitrary environments. These plans are concatenations of environmental constraint exploitations that ultimately lead to grasps.

I. INTRODUCTION

An *environmental constraint* (EC) is a feature of the environment that enables replacing aspects of control and/or perception with interaction between hand and environment. There is evidence that human grasping greatly benefits from the directed exploitation of environmental constraints. We have shown that humans increase their EC exploitation activity when visually impaired [4]. Another impressive case in favour of viewing grasping/manipulation as an intentional exploitation of ECs is this nineteen-second video segment[1].

Most recent advances in robotic grasping and manipulation can be viewed in the light of EC exploitation: Underactuated hand designs such as [5] and [3] benefit from the dynamics that occur between hand and environment instead of cancelling them out. Synergies [11] are another approach to concert hand motions based on the much lower-dimensional hand-environment interactions instead of explaining them by only considering the constraints imposed by the configuration space. The usefulness of EC exploitation immediately leads to the question of how to come up with the right ones and how to sequence them.

To plan EC exploitation, we must depart from state of the art approaches to grasp and manipulation planning, which require complete and precise models of the hand and the environment. The advantage of EC exploitation lies exactly in the elimination of this requirement, replacing it with simple sensing during EC exploitation. A planner for EC exploitation must be able to plan successfully in the absence of complete models, solely by using information directly perceivable from the environment.

The key insight towards the realization of these planners is the following: Each of the actions in a plan are themselves feedback controllers, capable of rejecting uncertainty and of acquiring valuable information about the state of the world. The planner must ensure that EC exploitation primitives are sequenced in such a way that they control the relevant world dimensions so as to ensure success.

A plan can be viewed as a composition of funnels, each of which represents an EC exploitation primitive. The planner

(a) *Increased Goal Detectability* means more options to chose from.

(b) *Increased Goal Reachability* means less uncertain actions.

(c) *Increased Goal Recognizability* means higher certainty about knowing whether an action successfully terminated.
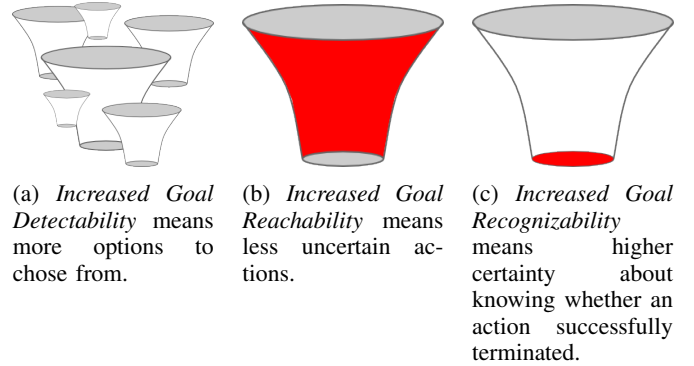
Fig. 1: Funnel analogy of exploiting environmental constraints.

must then sequence these funnels such that the EC exploitations ensure—due to their use of feedback based on EC—that the entrance to the next funnel (primitive) is reached when one primitive ends. Thus, we propose three reasons to exploit environmental constraints:

1) *Increased Goal Detectability*: ECs are easy to detect. These features, such as surfaces, edges, and corners, can be found reliably in a scene. In contrast to detecting objects, ECs are much less ambiguous in the function they offer and their detection need only few modalities (usually shape is enough).
2) *Increased Goal Reachability*: ECs reduce control requirements. Features such as surfaces reduce the DOFs of hand and object.
3) *Increased Goal Recognizability*: ECs are easy to recognize. These feature are per definition outstanding sensor events, such as the transition from non-contact to contact.

II. RELATED WORK

The idea of exploiting the environment and using contact to attain more robust motion plans is an old one [10]. Still, finding such fine motion plans grows double exponentially with the number of plan steps [2]. The geometric reasoning we apply is similar to the classical work on assembly planning [13]. The way we define pose constraints is similar to the task space regions [1] used for motion planning. The concept of contact-state graphs [8] is reflected in our representation of environmental constraints although spatial information also plays a significant role. Executing manipulation sequences based on termination predicates and online decisions are shown in [7]. Our work is complementary as it deals with how

to come up with these sequences. The task of sliding a book on a table and lifting it is formalized as trajectory optimization in [9]. We rather strive for a more general approach. Finally, in [12] the framework of optimal control is used to find manipulation actions that minimize uncertainty by contact with the environment. All of the mentioned approaches ignore the problem of modelling the environment from real sensor measurements. As will be shown, this will be a core part of our method.

III. SEQUENCES OF ENVIRONMENTAL CONSTRAINTS

We view the exploitation of a single environmental constraint as a funnel: Multiple world states are mapped onto a subset of them, thereby reducing the number of possible states. This simplifies the estimation and control problem. When sequencing two funnels we need to assess whether there is a set of states that is both part of the exit of the first funnel and part of the entry of the second funnel. Thus, our representation needs to be able to describe equivalent classes of states (due to environmental constraints) and to intersect between them.

Consequently, we define an environmental constraint as a set of spatial and contact constraints. It is described by the 6-tuple

$$s := (\{Hand, HandObject\},$$
$$T, e, O, F, \mathcal{T}),$$

where $Hand/HandObject$ signifies whether spatial and contact constraints are described between the hand and the environment only or the hand in combination with the object w.r.t. the environment. T is a homogeneous transformation describing the center of the spatial constraint and $e \in \mathbb{R}^3$ defines its extent in each dimension. Together they describe a bounding box of possible positions. The possible orientations are described by a finite set $O \subseteq SO(3)$. We descritize $SO(3)$ using 10860 equidistant rotations, each of which covers $\pm 6 \deg$ of angular displacement. Similar to the spatial constraints we represent the contact constraints by a set of forces F and torques $\mathcal{T}$. Examples of environmental constraint states are given in Sec. V.

We are finally looking for actions that exploit these environmental constraints. Here, actions are defined as controllers with desired spatial and contact profiles and a termination predicate that defines success:

$$a(s, s') = (u, term).$$

The termination predicate $term$ is defined as the intersection between (T, e, O) and (T', e', O') and the complement of $(F', \mathcal{T}')$ and $(F, \mathcal{T})$. If the intersection or the complement is empty there exists no single action that exploits environmental constraints to bring us from s to s'. Because of the complement operation actions are not symmetric, i.e. $a(s, s') \neq a(s', s)$. The desired control input u is fully specified by the intersection of the spatial constraints of s and s' and all constraints given by s.
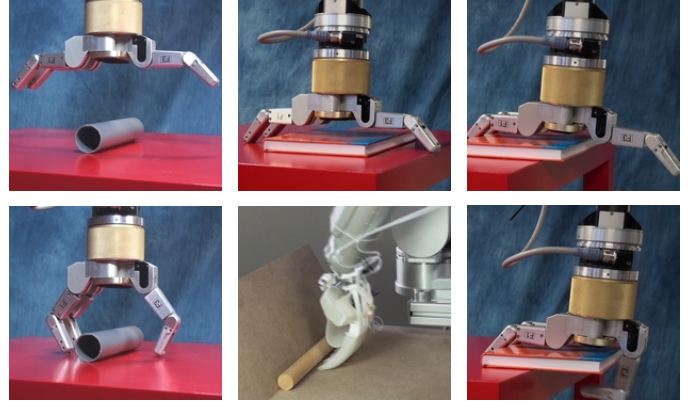

Fig. 2: **Top:** Visually-constrained positioning, surface-constrained caging, and surface-constrained sliding. **Bottom:** Surface-constrained, wall-constrained, and edge-constrained grasp.

To plan a sequence of actions that exploit environmental constraints and result in a grasp we propose the following procedure:

1) Environmental constraints according to the definition given above are extracted from a single 3D point cloud view of the scene. Each state is inserted as a node into a directed graph.
2) Actions between all pairs of environmental constraints are calculated. If such an action exists it is added as a directed edge between the corresponding nodes.
3) All grasp nodes are marked as goal nodes and the current robot/object state as a starting node. All paths that lead from start to goal are sequences of environmental constraints.

IV. ENVIRONMENTAL CONSTRAINT EXPLOITING ACTIONS

In the following we will depict three grasping strategies and three non-prehensile manipulations that make explicit use of environmental constraints.

Surface-Constrained Grasp: This grasping strategy can be applied whenever the object is placed on a flat support surface. The open hand approaches the object from above and establishes either palm-contact with the object itself or fingertip-contact with the support surface. During finger closing a compliant wrist position along the support surface normal guarantees slip along the environmental surface while contact locations between object and fingers remain stable. We detect surface-constrained grasps by fitting geometric primitives to the object shape which also determine the hand's pregrasp configuration [6].

Wall-Constrained Grasp: This grasping strategy exploits two surfaces that are perpendicular to each other. The open hand pushes the object along its support surface towards the second surface – the *wall*. While the object is caged between wall and palm, the fingers can slip underneath the object. We detect wall-constrained grasps by finding concave edges in the environment.

(a) Point cloud with object over-layed (inside red circle).

(b) The extracted visual positioning environmental constraint.

(c) Sliding constraint.

(d) Caging constraint.

(e) Surface-constrained box grasp.

(f) Edge grasp.

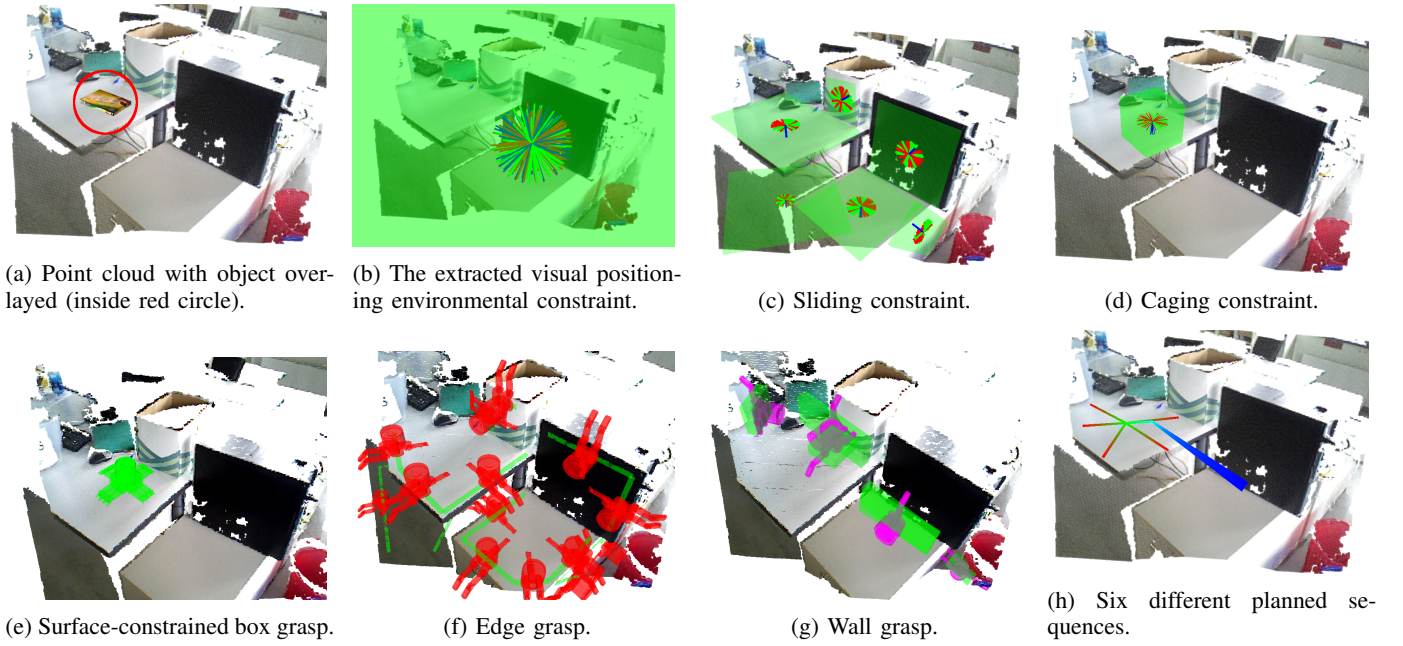(g) Wall grasp.

(h) Six different planned sequences.

Fig. 3: Office desk scene.

Edge-Constrained Grasp: Here, the object is assumed to be close to a convex edge. The open hand pushes the object over the edge and wraps its fingers around the newly exposed object surface. In contrast to the two strategies explained above, here the environment is not used to reduce the object's DOF during grasping. Instead the edge is a particular part of space that allows the hand to easily take over the DOF-constraining function of the environment.

Visually-Constrained Positioning: Inside the visible workspace the hand can be constraint visually, i.e. by visual servoing.

Surface-Constrained Caging: This action simply gets the hand/fingertips in contact with the environment. It can be applied whenever there is a small contiguous surface patch and the hand is close to it.

Surface-Constrained Sliding: This strategy assumes the hand is close to the object. During pushing the hand and object's DOFs are constrained by a support surface. Sliding constraints are recognized by extracting planar surfaces in the scene.

V. EXPERIMENTS

We obtained preliminary results of our method by applying it to an office desk scene depicted in Fig. 3a. The point cloud was recorded with an Asus Xtion Live depth sensor. Color information is not used at any stage of the algorithm. The recording does not contain the object to be grasped, rather we assume to have rough geometric knowledge about the entire object and a segmentation from the environment. Both, object and environment are assumed to be single rigid bodies. Fig. 3b-3d show the extracted environmental constraint states for the pregrasp-manipulation actions. The light green bounding boxes depict the position constraints (T, e), while

the orientation constraints are displayed by drawing 50 samples from O and plotting the RGB-axes. The pictures do not show the force/torque constraints $(F, \mathcal{T})$. Fig. 3e-3g display the found grasp states. Again, the light green boxes show the position constraints from which the grasp would succeed. The orientation constraints are here shown by plotting a 3D model of the Barrett hand BH-262.

In total 33 states that represent the six different kinds of environmental constraint exploitation were found in this scene. Finally, the algorithm found 6 different manipulation sequences that start the unconstrained robot hand and end in a grasp, see Fig. 3h. One ends in a surface-constrained box grasp, two in wall grasps, and three in edge grasp (the one on the left being a false positive). Not shown in the pictures, but extremely important for the execution of the plans are the termination predicates that depend on changes in the force/torque feedback. Additionally, we applied the algorithm to a scene with a box-like object lying in a shelf and on another office desk, see Fig. 4.
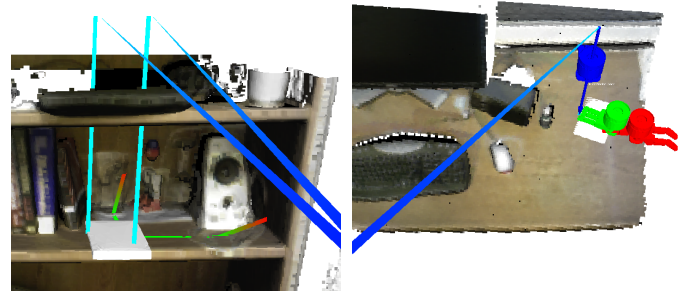

Fig. 4: Two other scenes: A shelf and another office desk.

VI. Conclusion

We presented a method to plan sequences of grasping strategies that exploit environmental constraints. We showed first promising results on real-world data and would like to strengthen those results by executing a plan on a robotic platform for the final workshop version.

References

[1] Dmitry Berenson, Siddhartha S. Srinivasa, and James Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, March 2011.

[2] John Canny. On computability of fine motion plans. pages 177–182 vol.1, May 1989.

[3] R. Deimel and O. Brock. A novel type of compliant, underactuated robotic hand for dexterous grasping. Berkeley, USA, 7 2014.

[4] Raphael Deimel, Clemens Eppner, Jos lvarez Ruiz, Marianne Maertens, and Oliver Brock. Exploitation of environmental constraints in human and robotic grasping. Singapore, Singapore, 12 2013.

[5] Aaron M. Dollar and Robert D. Howe. The highly adaptive SDM hand: Design and performance evaluation. 29(5):585–597, apr 2010.

[6] Clemens Eppner and Oliver Brock. Grasping unknown objects by exploiting shape adaptability and environmental constraints. 2013.

[7] Bernd Finkemeyer, Torsten Krger, and Friedrich M. Wahl. Executing assembly tasks specified by manipulation primitive nets. *Advanced Robotics*, 19(5):591611, 2005.

[8] Xuerong Ji and Jing Xiao. Automatic generation of high-level contact state space. *Int. J. Robotics Research*, 20: 238244, 1999.

[9] Jennifer King, Matthew Klingensmith, Christopher Dellin, Mehmet Dogar, Prasanna Velagapudi, Nancy Pollard, and Siddhartha Srinivasa. Pregrasp manipulation as trajectory optimization. In *Proceedings of Robotics Science and Systems (RSS)*, 2013.

[10] Tomás Lozano-Pérez, Matthew T. Mason, and Russell H. Taylor. Automatic synthesis of fine-motion strategies for robots. 3(1):3–24, March 1984.

[11] M. Santello, M. Flanders, and J.F. Soechting. Patterns of hand motion during grasping and the influence of sensory guidance. *The Journal of Neuroscience*, 22(4): 1426–1435, 2002.

[12] Marc Toussaint, Nathan Ratliff, Jeannette Bohg, Ludovic Righetti, Peter Englert, and Stefan Schaal. Dual execution of optimized contact interaction trajectories. 2014.

[13] Randall H. Wilson and Jean-claude Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71:371396, 1994.