A Constraint-Aware Motion Planning Algorithm for Robotic Folding of Clothes

Apoorva Sachdev

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-92
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-92.html

May 11, 2012

Acknowledgement

I would like to express my sincere gratitude to my adviser Professor Pieter Abbeel for his continuous support and encouragement. I would like to thank Dr.Dmitry Berenson for guiding and mentoring us throughout the project. My sincere thanks also goes to Professor Ken Goldberg for his insightful comments. I thank my fellow labmates, Karthik Lakshmanan and Ziang Xie for the numerous sleepless nights we spent working together, without you guys this would have been impossible. Last but not the least, I would like to thank my family and friends, for being so supportive and for always being there for me!

A Constraint-Aware Motion Planning Algorithm for Robotic Folding of Clothes

Karthik Lakshmanan, Apoorva Sachdev, Ziang Xie, Dmitry Berenson, Ken Goldberg, and Pieter Abbeel

1 Motivation, Problem Statement, Related Work

Robotic manipulation of 2D deformable objects is a difficult problem largely because such objects typically have infinite-dimensional configuration spaces and are too computationally expensive to simulate in the inner-loop of a motion planner.

The problem we address is as follows: Given a robot model, the shape of a piece of cloth in a spread-out configuration on a horizontal table, and a final folded configuration specified by a sequence of g-folds, output a sequence of robot motions that achieve the final folded configuration or report that none exists.

The state-of-the-art approach is g-folding [21]. At the core of this approach is the definition of a cloth model that allows reasoning about the geometry rather than the physics of the cloth in relevant parts of the state space. Given the geometry of the cloth, their algorithm computes how many grippers are needed and prescribes motions for these grippers to achieve the final configuration, specified as a sequence of g-folds—folds that can be achieved while staying in the subset of the state space to which their geometric model applies. G-folding, however, has severe practical limitations: due to robot and environmental constraints, the gripper motions produced by g-folding are often infeasible. When that happens, the top-down approach followed by g-folding fails.

Our work builds on g-folding [21], but does account for kinematic restrictions posed by real robots. Our work also draws from the work of Bell and Balkcom [4, 5], which deals with computing the grasp points needed to immobilize a polygonal non-stretchable piece of cloth. Relevant work in cloth folding includes dynamic towel folding [1], cloth perception [18], and strategies to bring the cloth into a spread-out configuration where folds can be applied [17, 10, 6]. Our work is also related to work in motion planning that searches over primitives to construct robot trajectories [12, 7, 16, 20, 9, 19, 14]. Other relevant work has been done in physical simulation of cloth [3, 8, 11], origami folding [2], carton folding [15, 11], and metal bending [13], which use similar material models to the one presented here.

2 Technical Approach

We have developed a motion planning approach that plans directly at the level of robotic primitives rather than the gripper motions used in [21]. Given a sequence of g-folds that take the cloth from the initial to the final configuration, our approach determines whether a sequence of motion primitives exists that results in the success-

UC Berkeley

ful execution of all specified g-folds. If so, the algorithm outputs the robot motion which brings the cloth to the final configuration.

Our approach is based on a search space formulation which only allows actions the robot can execute from the current robot and clothing configuration. To restrict the (otherwise unmanageably large) search space we define a class of motion primitives and search over sequences of these primitives to perform the required folds. Our class of primitives consists of all motions that begin and end with the cloth in a *g-state* — a configuration of the cloth where all parts of the cloth are either horizontal or vertical. This is a broader class of motion than was allowed in [21] because the intermediate states of the cloth during execution of primitives are unrestricted.

With this search space formulation in place, we can associate costs with each g-primitive and search for optimal solutions. In our experiments the cost we associated with each g-primitive is the time it takes to execute, which is readily computed from the length of the motion performed by the robot. This cost is specific to each instantiation of a g-primitive (rather than a single fixed cost being associated with, for instance, a g-fold). Hence not only does our approach allow us to find solutions in cases where g-folding would simply fail, it also enables finding better (i.e., time-optimal) solutions.

We use a search-based procedure to find a sequence of folding primitives that results in the successful completion of all the requested g-folds, if possible. Our planning algorithm is comprised of two components: (1) The creation of a Fold-DAG and (2) A search over motion primitives. We use a directed acyclic graph (DAG), which we call the Fold-DAG, to capture dependencies between the folds specifed in a folding sequence, i.e. which g-folds need to be completed before a particular g-fold can be performed, this allows us the freedom to sequence folds differently from the sequence specified by the user and potentialy fold the article faster than the input sequence allows.

Given vertices of the cloth, the robot model, and the Fold-DAG we can compute a time-optimal motion sequence of primitives for the robot to execute in order to reach the desired final configuration using the A* search algorithm. For the purposes of A*, we define a goal state as a state which has no g-folds remaining in the DAG. For a given state, we generate the successors by applying all the primitives in our set of primitives. If a given primitive is infeasible (due to, for instance, reachability contraints), it does not generate a successor. Since we want the search algorithm to return the plan of least execution time, the path-cost to reach a state from the starting state is a measure of the time taken by the robot to perform all primitives that constitute the path. Our A* heuristic uses a relaxation of the problem which ignores robot constraints and approximates the time taken for the robot to perform all remaining folds in the DAG at the given state. This is computed as the sum of the time taken for the grippers to traverse the straight line distance between grip locations and endpoints for each fold assuming constant arm speed. This straight line approximation is an underestimate of time needed to execute the fold trajectory with the robot, making the heuristic admissible.

In order to make the search process computationally feasible, we must ensure that the search space we consider is not unmanagably large. Allowing any arbitrary motion to be considered would cover the state space, but it would also produce an unmanagable search space and would require simulations of the cloth dynamics (which are computationally expensive and sometimes inaccurate). We thus restrict the search to a class of motion primitives we call *g-primitives*. Intuitively, a g-primitive is any motion primitive that both starts and ends with the system in a g-state. The primitives used in our experiments are of three types: performing a g-fold (including folds that allow the article to hang from the table), dragging the article along the table, and base motion.

3 Planning Results

We used a Willow Garage PR2 robotic platform [15] and performed experiments both in simulation and on the physical robot. We experimented with various clothing articles like towels, t-shirts, long-sleeved shirts and pants. The folding sequence is given as input to the algorithm. To evaluate the efficiency of our planning approach we ran tests on several articles in simulation and computed the time necessary to find a solution and the time taken by different components of the algorithm. An example input fold sequence, Fold-DAG, and plan are shown in Figure 1.

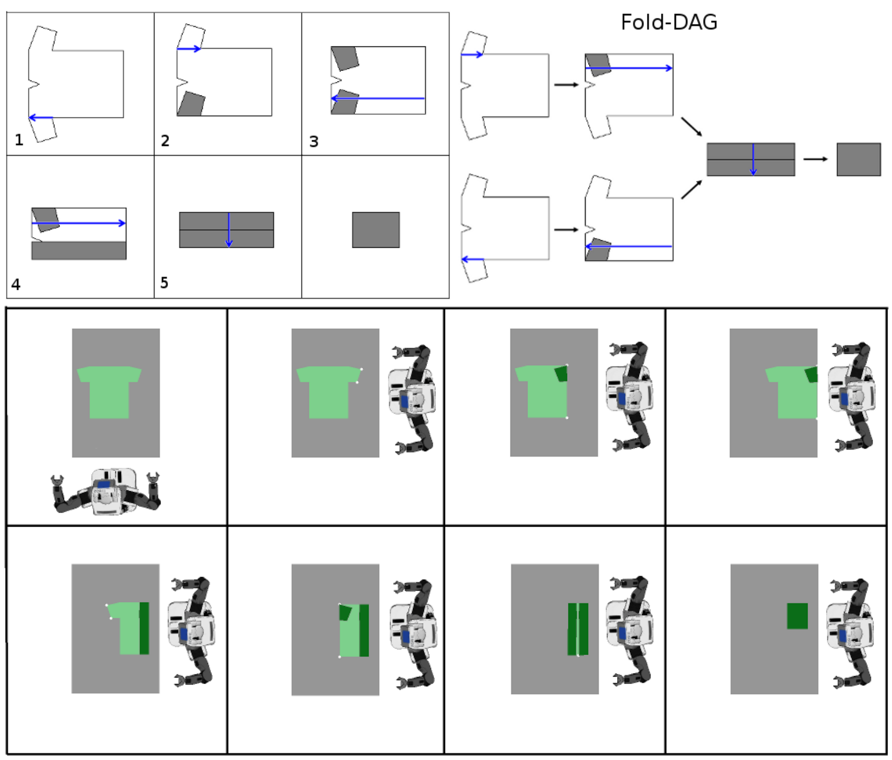

Fig. 1: T-shirt input folds and solution. Upper left: User input folds. Upper right: Automatically-generated Fold-DAG. Bottom: Plan generated by A*.

Article	**# Folds**	**L**	**C**	**N Expanded**	**N Explored**	**T**	**IK time**	**IK Calls**	**Overhead**
Tshirt	5	7	61.2s	101	614	68.0s	23.2s	3512	0.03s
Jeans	2	4	33.9s	27	211	17.8s	6.4s	776	0.015s
Shirt	7	9	76.8s	210	1149	152.9s	46.0s	5911	0.067s
Tie	3	4	26.8s	21	217	31.2s	4.7s	593	0.005s
Scarf	2	3	25.2s	12	128	10.5s	4.4s	593	0.03s
Vest	2	3	30.1s	23	142	9.2s	2.6s	383	0.006s
Skirt	3	5	47.7s	380	2715	97.47s	89.8s	10087	0.15s
Big Towel	3	6	46.3s	253	1983	127.0s	61.8s	6675	0.09s
Hand Towel	3	3	25.8s	13	105	8.5s	3.7s	400	0.003s

Table 1: Simulation Results. L is the number of primitives in the solution path, C is the cost of the path in seconds. N expanded/explored is the number of nodes expanded/explored. T is the total search time of algorithm.

We used the ikfast module provided by OpenRAVE in order to determine if the robot can reach a given point. In order to make the generated plans robust to robot execution error (for example, dragging by less than the desired amount), we introduce the concept of "IK comfort". We only declare a point reachable if both the point and four of points on the circumference of a circle of a set radius, centered at the given point are reachable. Also, if the gripper fails to grab the cloth at a particular point, it tries to grab other points that lie within the comfort radius of the point before failing. For our trials, we found that a comfort radius of 3cm resulted in robust execution.

Article	**# Folds**	**L**	**C**	**N Expanded**	**N Explored**	**T**	**IK time**	**IK Calls**	**Overhead**
Tshirt	5	7	61.2s	101	614	82.5s	26.1s	3482	0.04s
Jeans	2	4	33.9s	27	206	40.5s	29.4s	2934	0.007s
Shirt	7	9	76.8s	209	1133	320.9s	217.1s	23214	0.08s
Tie	3	4	26.8s	21	217	38.8s	24.5s	2582	0.014s
Scarf	2	3	25.2s	12	128	28.1s	22.5s	2344	0.004s
Vest	2	3	30.1s	23	139	9.2s	21.4s	1915	0.006s
Skirt	3	6	43.86s	287	2167	336.5s	264.6s	31894	0.1235s
Big Towel	3	6	46.8s	251	1834	276.6s	198.1s	21447	0.1159s

Table 2: Simulation Results with IK comfort of 3 cm. See symbol definitions in Table 1.

4 Experimental Results

We used a rectangular table with a soft working surface, so that the relatively thick grippers can easily get underneath the cloth. At the beginning, the robot can always see the entire clothing article in a known, fully spread-out configuration.

For several of the articles, we executed the generated plans multiple times on the robot. Table 3 shows the results of our runs. Several snapshots from folding a

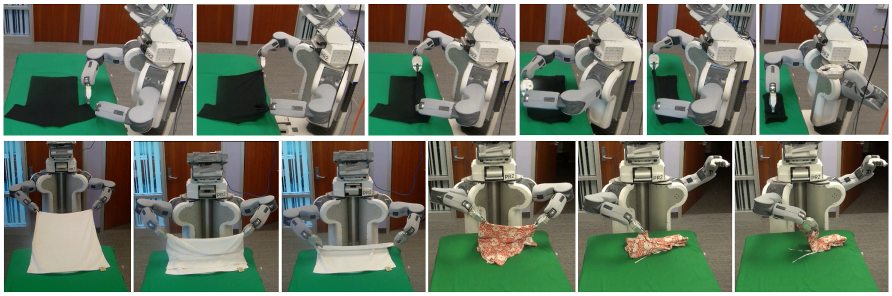

Fig. 2: T-shirt, towel and skirt folding sequences executed by PR2.

Clothing item	Success Rate	Average Execution Time
Tshirt	3/4	216.5s
Large towel	2/4	102s
Hand towel	4/4	75s
Jeans	4/5	108.5s
Skirt	5/5	82.5s

Table 3: Success rate and execution time of physical robot execution.

T-shirt, towel, and skirt are shown in Figure 2. Videos of the executions are posted at `http://rll.berkeley.edu/iser2012-folding`.

Our current system does not use visual feedback during execution. In order to close the loop, we simulate visual feedback using a human in the loop system. At the beginning of each primitive, the GUI highlights grip points on the polygon. A human then clicks the corresponding points in the live stereo camera feed. The end points are then translated by the error between the expected and the clicked grip points, and the gripper trajectory is recomputed using the new start and end points.

5 Main Experimental Insights

As illustrated by our success rates on the various clothing articles, our method shows a high level of reliability on real cloth, even if it does not perfectly conform to our assumptions. For example, jeans and large towels clearly violate the zero thickness assumption, while the frills and pleats of the skirt are not taken into consideration by our cloth model. However, we are able to achieve high success rates on both these articles.

Our failures typically arise from errors in robot execution, particularly base motion. If the base does not move by the desired amount, a grip point might become unreachable. The introduction of IK comfort along with making the arms correct for base motion undershoot during a drag greatly reduces the number of such failures.

Our experiments show that the planned execution times typically underestimate the real execution times observed with the PR2. This is because the costs for each primitive used in the planning phase are highly idealized. A large part of this discrepancy can be attributed to the Move primitive. The 2D navigation package on board the PR2 causes it to detect false obstacles at times. This results in the robot stopping multiple times during the move. The planner also ignores certain other behaviors of robot execution. For example, the PR2 may fail to grab the cloth on the first attempt, and would need to move the gripper to regrasp the cloth. Additionally, the planner assumes a constant velocity for base movement, while the robot actually spends more time accelerating and decelerating than at its full speed.

In conclusion, we described a motion planning algorithm for robotic cloth folding, enabling us to avoid computationally expensive physics simulations while taking into account kinematic constraints. We presented examples of cloth manipulation primitives that allow the robot to perform a set of user defined g-folds using our simplified cloth model. Our search algorithm allowed the robot to choose a sequence of primitives to perform all given folds in the shortest possible time (given the available primitives). At the core of our method is the consideration of real robot limitations. Our experiments show that (a) many articles of clothing conform well enough given the assumptions made in our model and (b) this approach allows our robot to perform a wide variety of folds on articles of various sizes and shapes.

References

1. B. Balaguer and S. Carpin. Combining imitation and reinforcement learning to fold deformable planar objects. *IROS*, 2011.
2. D. Balkcom and M. Mason. Robotic origami folding. *IJRR*, 27(5):613–627, 2008.
3. D. Baraff and A. Witkin. Large steps in cloth simulation. *SIGGRAPH*, 1998.
4. M. Bell. Flexible object manipulation. *PhD Thesis, Dartmouth College*, 2010.
5. M. Bell and D. Balkcom. Grasping non-stretchable cloth polygons. *IJRR*, 29:775–784, 2010.
6. Christian Bersch, Benjamin Pitzer, and Soren Kammel. Bimanual robotic cloth manipulation for laundry folding. In *IROS*, pages 1413 –1419, Sept. 2011.
7. Michael S. Branicky, Ross A. Knepper, and James J. Kuffner. Path and trajectory diversity: Theory and algorithms. In *ICRA*, 2008.
8. R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact, and friction for cloth animation. *SIGGRAPH*, 2002.
9. B. J. Cohen, S. Chitta, and M. Likhachev. Search-based planning for manipulation with motion primitives. In *ICRA*, pages 2902 –2908, May 2010.
10. Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James O'Brien, and Pieter Abbeel. Bringing clothing into desired configurations with limited perception. In *ICRA*, 2011.
11. N. Fahantidis, K. Paraschidis, V. Petridis, Z. Doulgeri, L. Petrou, and G. Hasapis. Robot handling of flat textile materials. *IEEE Robotics and Automation Magazine*, 4(1):34–41, 1997.
12. C. Green and Alonzo Kelly. Toward optimal sampling in the space of paths. In *ISRR*, 2007.
13. S. Gupta, D. Bourne, K. Kim, and S. Krishnan. Automated process planning for robotic sheet metal bending operations. *Journal of Manufacturing Systems*, 17:338–360, 1998.
14. Kris Hauser, Tim Bretl, Kensuke Harada, and Jean-claude Latombe. Using motion primitives in probabilistic sample-based planning for humanoid robots. In *WAFR*, 2006.
15. T. Kanade. A theory of origami world. *Artificial Intelligence*, 13(3):279–311, 1980.
16. R. Knepper and M. Mason. Empirical sampling of path sets for local area motion planning. In *ISER*. Springer, 2008.
17. J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *ICRA*, 2010.
18. S. Miller, M. Fritz, T. Darrell, and P. Abbeel. Parametrized shape models for clothing. *ICRA*, 2011.
19. P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. learning and generalization of motor skills by learning from demonstration. In *ICRA*, 2009.
20. Mihail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Mobile Robot Motion Planning. *Journal of Field Robotics*, 26(3):308–333, 2009.
21. J. van den Berg, S. Miller, K. Goldberg, and P. Abbeel. Gravity-based robotic cloth folding. *WAFR*, 2010.