Gravity-Based Robotic Cloth Folding

Jur van den Berg, Stephen Miller, Ken Goldberg, Pieter Abbeel

Abstract We consider how the tedious chore of folding clothes can be performed by a robot. At the core of our approach is the definition of a cloth model that allows us to reason about the geometry rather than the physics of the cloth in significant parts of the state space. We present an algorithm that, given the geometry of the cloth, computes how many grippers are needed and what the motion of these grippers are to achieve a 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 the geometric model applies. G-folds are easy to specify and are sufficiently rich to capture most common cloth folding procedures. We consider folds involving single and stacked layers of material and describe experiments folding towels, shirts, sweaters, and slacks with a Willow Garage PR2 robot.

1 Introduction

An English patent for a clothes washing machine was issued in 1691. Since then, there have been many innovations in washing and drying, but folding of clothes remains a manual (and notoriously tedious) activity. In this paper, we present a geometric model and algorithms that are steps toward autonomous robot folding of clothes. Cloth is highly non-rigid, flexible, and deformable with an infinite-dimensional configuration space. We consider articles of clothing that can be described by a simple polygonal boundary when lying flat on a horizontal surface.

We introduce a deterministic geometric model of cloth motion based on gravity and assumptions about material properties. We constrain robot motion such that at all times, one part of the cloth is lying horizontally on the table and one part (possibly empty) hangs vertically from the grippers parallel to the gravity vector. This allows a configuration of the cloth to be fully determined by the line that separates

University of California, Berkeley. E-mail: {berg, sdavidmiller, goldberg, pabbeel}@berkeley.edu

the horizontal and the vertical parts. We call this line the *baseline*, which defines a 2-D configuration space for the material.

Given polygonal geometry of the cloth, number of grippers, and desired fold sequence (see Fig. 1), we present an algorithm that computes a motion plan for the grippers that moves the cloth through the C-space to reach the desired final arrangement, or a report that no such motion plan exists. We implemented the algorithm on a Willow Garage PR-2 robot and report experiments folding towels, t-shirts, sweaters, and slacks.

The remainder of this paper is organized as follows. In Section 2 we discuss related work. In Section 3 we define the problem. In Section 4 we describe our algorithm to compute the manipulation motion of the robot to execute a given folding sequence using g-folds. In Section 5 we report experimental results folding towels, t-shirts, and slacks using a Willow Garage PR-2 robot. We conclude in Section 6.

2 Related Work

The work that is most related to ours is the work of Bell and Balkcom [3, 4]. In [4], the grasp points are computed to immobilize a polygonal non-stretchable piece of cloth. Gravity is used to reduce the number of grasp points to hold cloth in a predictable configuration, potentially with a single fold, using two grippers in [3]. We extend this work and include a folding surface. We assume that points that are lying on a table are fixed by friction and gravity, and need not be grasped. The work of [3] also shows how to fold a t-shirt using the *Japanese method*; the fold can be achieved by grasping the cloth at three points without regrasping.

In [7], the robot handling of cloth material is discussed and some specific folds are shown. The work of [14] also discusses a specific folding manipulation. The work of [12] deals specifically with folding towels. The work focuses on visual detection of the vertices of the towel, and use a scripted motion to achieve folds using a PR-2 robot. We build on the results of this work in our experiments.

There is also quite a large body of work on *cloth simulation*, which simulates the behavior of cloth under manipulation forces using the laws of physics [2, 5, 6]. In our work, we manipulate cloth such that it is always in a configuration that allows us to reason about the geometry of the cloth, rather than about its physics and dynamics.

Folding has been extensively studied in the context of *origami* [1, 9, 10]. Origami, or paper folding, is fundamentally different from cloth folding, since unfolded regions of the paper are considered to be rigid facets connected by “hinges” that model the creases in the paper. In contrast, cloth material is flexible everywhere. Yet, we draw from results in paper folding in our work. Applications of paper folding outside origami include box folding [13, 11] and metal bending [8], where the material model is essentially the same as that of paper.

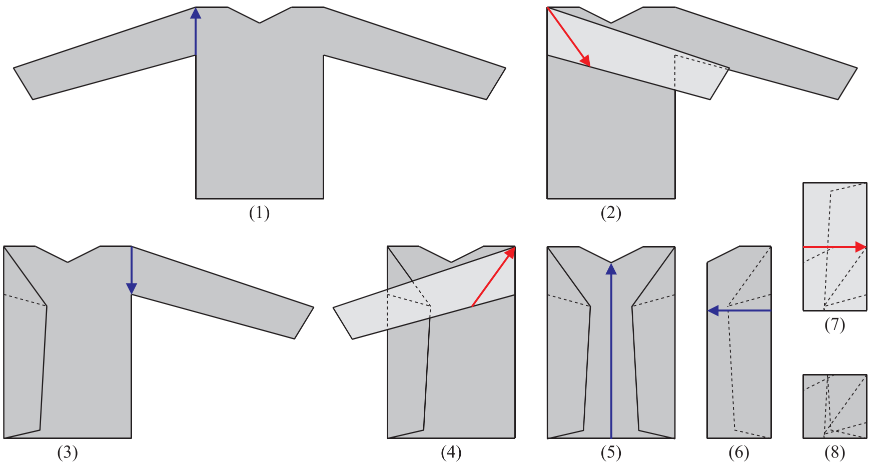


Fig. 1 Folding a long-sleeve into a square using a sequence of seven g-folds. Red g-folds apply to the geometry that was folded in the preceding g-fold. Blue g-folds apply to the entire geometry.

3 Problem Description

We assume gravity acting in the downward vertical ($-z$) direction and a sufficiently large planar table in the horizontal (xy) plane. We assume the article of clothing can be fully described by a simple *polygon* (convex or non-convex) initially lying on the horizontal surface. We are given the initial n vertices of the polygonal cloth in counterclockwise order.

We make the following standard assumptions on the cloth material:

1. The cloth has *infinite flexibility*. There is no energy contribution from bending.
2. The cloth is *non-stretchable*. No geodesic path lengths can be increased.
3. The cloth has *infinite friction* with the surface on which it lies and with itself.
4. The cloth has *zero thickness*.
5. The cloth is *subject to gravity*.
6. The cloth has *no dynamics*.

At the core of our approach is the following additional assumption, which we call the *downward tendency assumption:*

7. If the cloth is held by a number of grippers, and one or more grippers release the cloth, no point of the cloth will move upwards as a result of gravity and internal forces within the cloth.

This assumption does not directly follow from physics, rather it is an approximation which seems to match the behavior of reasonably shaped cloth, such as everyday clothing articles, surprisingly well[1], and it allows us to reason purely about the *geometry* rather than the physics of the cloth.

[1] The assumption is not accurate for an exotic family of shapes called pinwheels, as shown in [3].

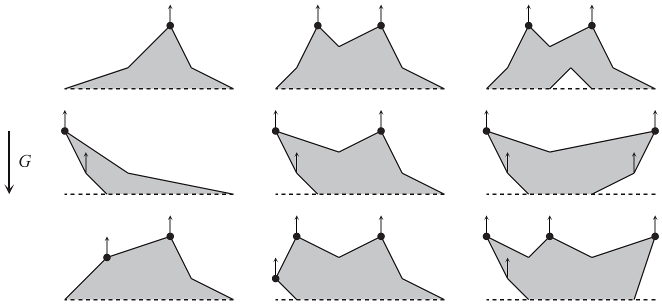


Fig. 2 Examples of vertical parts of cloths in various configurations. In order for the cloth not to be immobilized, all convex vertices not at the baseline at which the negative gravity vector (small arrows) does not point into the cloth must be grasped. These vertices are indicated by the dots.

The downward-tendency assumption allows the cloth to be held by the grippers such that one section lies horizontally on the surface and another section hangs vertically. The line that separates the horizontal and the vertical parts is called the *baseline*. To ensure deterministic behavior of the cloth, the grippers must be arranged so that the vertical section is immobilized. The points that are lying on the surface (including those on the baseline) are immobilized, as they cannot move in the plane due to friction and will not move upward per the downward-tendency assumption, so they need not be grasped. Fig. 2 shows an example, where points of the cloth are held by grippers. To make sure that the vertical part of the cloth is immobilized, it turns out that every *convex* vertex of the vertical part of the cloth at which the negative gravity vector does not point into the cloth polygon must either be held by a gripper or be part of the baseline. This follows from the following theorem:

Theorem 1. *In our material model, a vertically hanging cloth polygon is immobilized when every* convex *vertex of the cloth at which the negative gravity vector does not point into the cloth polygon is fixed (i.e. be held by a gripper or be part of the baseline).*

Proof: By [3], we know that a non-stretchable planar tree is fully immobilized if each node of the tree of which its incident edges do not positively span $\mathbb{R}^2$ is fixed. Now, let us define an *upper string* of a polygon as a maximal sequence of edges of which the extreme vertices are convex vertices of the polygon, and no part of the polygon lies above the edges (see Fig. 3(a)). A given polygon P can have multiple upper strings, but has at least one.

For each upper string holds that at its convex vertices the negative gravity vector points outside the polygon. As these convex vertices are fixed (by a gripper), the entire set of edges the string consists of is immobilized. This can be seen by adding virtual vertical edges fixed in gravity pointing downward from the non-convex vertices, which make sure that the non-convex vertices cannot move upward (per the downward-tendency assumption). The incident edges of the non-convex vertices now positively-span $\mathbb{R}^2$, hence the entire string is immobilized.

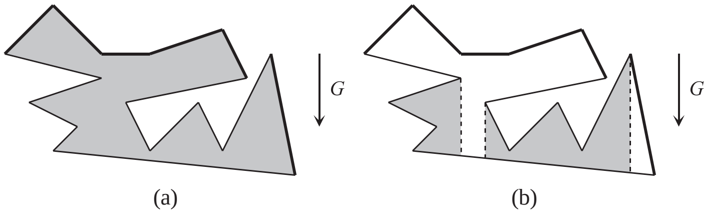


Fig. 3 **(a)** A polygon with two upper strings shown thick. **(b)** The white part of the polygon (including the vertical dashed edges) has proven immobilized. The grey part remains.

Now, every point of the polygon P that can be connected to an upper string by a vertical line segment that is fully contained within P is immobilized. This is because this point cannot move downward per the non-stretchability assumption (note that the upper string is immobilized), and it cannot move upward per the downward-tendency assumption. Hence, all such points can be "removed" from P – they have been proven immobilized. What remains is a smaller polygon P' (potentially consisting of multiple pieces) for which immobilization has not been proven (see Fig. 3(b)). The smaller polygon P' has vertical edges that did not belong to the original polygon P. The points on these vertical edges are immobilized, including both incident vertices (of which the upper one may be a non-convex vertex of P that is convex in P'), as they vertically connect to the upper string.

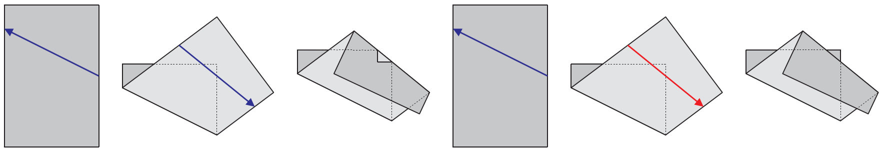
Then, the proof recurses on the new polygon P', of which the convex vertices of the upper string(s) need to be fixed. Note that P' may have convex vertices that were non-convex in P. These need not be fixed, as they were already proven immobilized since they are part of the vertical edge of P'.

This proves the theorem. Note that convex vertices where the negative gravity vector points into the polygon will never be part of an upper string at any phase of the proof, so they need not be fixed. Also, the recursion "terminates." This can be seen by considering the vertical trapezoidal decomposition of the original polygon P, which contains a finite number of trapezoids. In each recursion step, at least one trapezoid is removed from P, until the entire polygon has proven immobilized. □

A *g-fold* (g refers to gravity) is specified by a directed line segment in the plane whose endpoints lie on the boundary of the cloth polygon. The segment partitions the polygon into two parts, one to be folded over another (see Fig. 4). A g-fold is successfully achieved when the part of the polygon to the *left* of the directed line segment is folded across the line segment and placed horizontally on top of the other part, while maintaining the following property:

- At all times during a folding procedure, every part of the cloth is either horizontal or vertical, and the grippers hold points on the vertical part such that it is immobilized (see Fig. 5).

This ensures that the cloth is in a fully predictable configuration according to our material model at all times during the folding procedure. Not all folds can be



(a) (b)

Fig. 4 **(a)** A g-fold is specified by a directed line segment partitioning the (stacked) geometry into two parts. The g-fold is successfully achieved when the part of the geometry left of the line segment is folded around the line segment. A sequence of two g-folds is shown here. **(b)** A g-fold sequence similar to (a), but the second g-fold (a red g-fold) is specified such that it only applies to the part of the cloth that was folded in the previous g-fold.

achieved using a g-fold; in terms of [1], *valley folds* can be achieved using a g-fold, but *mountain folds* cannot.

A *g-fold sequence* is a sequence of g-folds as illustrated in Fig. 1. After the initial g-fold, the *stacked* geometry of cloth allows us to specify two types of g-fold: a "red" g-fold and a "blue" g-fold. A blue g-fold is specified by a line segment partitioning the polygon formed by the *silhouette* of the stacked geometry into two parts, and is successfully achieved by folding the (entire) geometry left of the line segment. A red g-fold is similarly specified, but only applies to the (potentially stacked) geometry that was folded in the previous g-fold (see Fig. 4).

We are given a robot with k point grippers that can grasp the cloth at any point on the *boundary* of the polygon formed by the silhouette of the stacked geometry. At each such point, the gripper will grasp all layers of the stack at that point (i.e., it is not capable of distinguishing between layers). Each of the grippers is able to move independently above the xy-plane and we assume that gripper motion is exact.

The problem we discuss in this paper is then defined as follows. Given a specification of a sequence of g-folds, determine whether each of the folds are feasible given the number of grippers available, and if so, compute the number of grippers needed and the manipulation motion for each of the grippers to achieve the g-folds.

4 Planning G-folds

4.1 *Single G-folds on Unstacked Geometry*

We first discuss the case of performing a single g-fold of the original (unstacked) polygon. During the manipulation, the cloth must be separated in a vertical part and a horizontal part at all times. The line separating the vertical part and the horizontal part is called the *baseline*.

Given a polygonal cloth and a specification of a g-fold by a directed line segment (e.g. the first g-fold of Fig. 4(a)), we plan the manipulation as follows. The manipulation consists of two phases: in the first phase, the cloth is manipulated such that

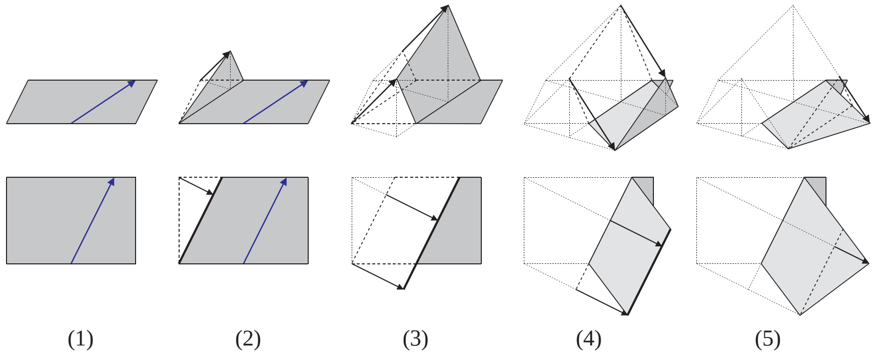


Fig. 5 The motion of two grippers (arrows) successfully performing the first g-fold specified in Fig. 4(a) shown both in 3-D view and top-view. At all times during the manipulation, all parts of the cloth are either vertical or horizontal and immobilized. The boundary between the vertical part and the horizontal part of the cloth is called the *baseline*.

the part that needs to be folded is brought vertical above the line segment specifying the g-fold (see Figs. 5-(1), 5-(2), and 5-(3)). In the second phase, the g-fold is completed by manipulating the cloth such that the vertical part is laid down on the surface with its original normal reversed (Figs. 5-(3), 5-(4), and 5-(5)).

The first phase is carried out as shown in Figs. 5-(1), 5-(2) and 5-(3), manipulating the cloth such that the baseline of the vertical part is parallel to the line segment at all times. Initially, the "baseline" is outside the cloth polygon (meaning that there is no vertical part) and is moved linearly towards the line segment specifying the g-fold. In the second phase, the g-fold is completed by laying down the vertical part of the cloth using a mirrored manipulation in which the baseline is again parallel to the line segment at all times. Initially the baseline is at the line segment specifying the g-fold and in moved linearly outward until the baseline is outside the folded part of the polygon (see Figs. 5-(3), 5-(4) and 5-(5)).

The corresponding motions of the grippers holding the vertices can be computed as follows. Let us assume without loss of generality that the line segment specifying the g-fold coincides with the x-axis and points in the positive x-direction. Hence, the part of the polygon above the x-axis needs to be folded. Each convex vertex of this part in which the positive y-vector points outside of the cloth in its initial configuration needs to be held by a gripper at some point during the manipulation. We denote this set of vertices by V and can be computed in $O(n)$ time. Let y^* be the maximum of the y-coordinates of the vertices in V. Now, we let the baseline, which is parallel to the x-axis at all times, move "down" with speed 1, starting at $y_b = y^*$, where y_b denotes the y-coordinate of the baseline. Let the initial planar coordinates of a vertex $v \in V$ be (x_v, y_v). As soon the baseline passes y_v, vertex v starts to be manipulated. When the baseline passes $-y_v$, vertex v stops to be manipulated. During the manipulation, the vertex is held precisely above the baseline. In general, the 3-D coordinate $(x(y_b), y(y_b), z(y_b))$ of the gripper holding vertex v as a function of the y-coordinate of the baseline is:

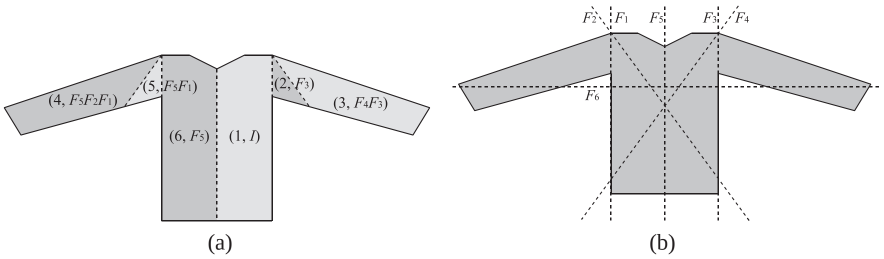


Fig. 6 (a) The representation of folded stacked geometry. The example shown here is the longsleeve t-shirt of Fig. 1 after five g-folds. With each facet, the stack height (integer) and a transformation matrix is stored. **(b)** Each transformation matrix F_i corresponds to mirroring the geometry in the line segment specifying the i'th g-fold.

$$x(y_b) = x_v, \qquad y(y_b) = y_b, \qquad z(y_b) = y_v - |y_b|, \tag{1}$$

for $y_b \in [y_v, -y_v]$. Outside of this interval, the vertex is part of the horizontal part of the cloth and does not need to be grasped by a gripper. This reasoning applies to all vertices $v \in V$. When the baseline has reached $-y^*$, all vertices have been laid down and the g-fold is completed. As a result, we do not need to grasp any vertex outside of V at any point during the manipulation, and the motions of the vertices in V can be computed in $O(n)$ time.

4.2 Sequences of G-folds and Stacked Geometry

We now discuss the case of folding already folded geometry. First, we discuss how to represent folded, stacked geometry. Let us look at the example of the longsleeve t-shirt of Fig. 1, and in particular at the geometry of the cloth after five g-folds. The creases of the folds have subdivided the original polygon into facets (see Fig. 6(a)). With each such facet, we maintain two values: an integer indicating the height of the facet in the stacked geometry (1 is the lowest) and a transformation matrix indicating how the facet is transformed from the original geometry to the folded geometry. Each transformation matrix is a product of a subset of the matrices F_i that each correspond to the mirroring in the line segment specifying the i'th g-fold. In Fig. 6(b), we show the lines of each of the g-folds with the associated matrix F_i.

Given the representation of the current stacked geometry and a line segment specifying a new g-fold, we show how we manipulate the cloth to successfully perform the g-fold or report that the g-fold is infeasible. We assume that the line segment specifying the g-fold partitions the silhouette of the stacked geometry into two parts (i.e., a blue g-fold). Let us look at the sixth specified g-fold in the longsleeve t-shirt example, which folds the geometry of Fig. 6.

Each facet of the geometry (in its folded configuration) is either fully to the left of the line segment, fully to the right, or intersected by the line segment specifying

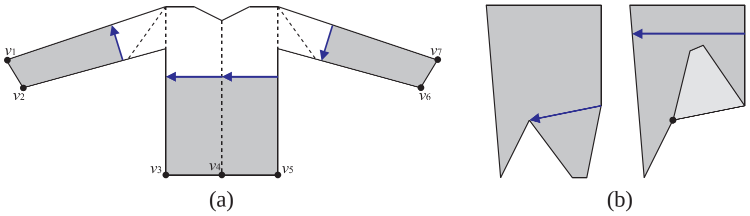


Fig. 7 **(a)** The geometry in the longsleeve t-shirt example after subdividing the facets by the line segment specifying the sixth g-fold. The gray facets need to be folded. The convex vertices for which the negative gravity vector points outside of the facet are shown using dots. **(b)** The vertex marked by the dot need not be held by a gripper to perform the fold, as it is non-convex in the dark gray facet (even though it is convex in the light gray facet).

the g-fold. The facets intersected by the line segment are subdivided into two new facets, both initially borrowing the data (the stack height and the transformation matrix) of the original facet. Now, each facet will either be folded, or will not be folded. Fig. 7(a) shows the new geometry in the longsleeve t-shirt example after subdividing the facets by the line segment specifying the g-fold. The gray facets need to be folded.

As in the case of folding planar geometry, for each facet each convex vertex at which the gravity vector points outside of the facet at the time it is above the line segment specifying the g-fold should be held by a gripper, and each non-convex vertex and each convex vertex at which the negative gravity vector points inside the facet need not be held by a gripper. If multiple facets share a vertex, and according to at least one facet it needs not be held by a gripper, it does not need to be held by a gripper (see Fig. 7(b)).

For the t-shirt example, the vertices that need to be grasped are shown using dots in Fig. 7(a) and labeled $v_1, \dots, v_7$. Applying the transformation matrices stored with the incident facet to each of the vertices shows that v_1, v_3, v_5, and v_7 will coincide in the plane. As a gripper will grasp all layers the geometry, only one gripper is necessary to hold these vertices. Vertex v_4 also needs to be held by gripper. Vertices v_2 and v_6 remain, but they need *not* be grasped. This is for the following reason. As can be seen in Fig. 1, these vertices are fully *covered*. That is, the vertex is "hidden" behind other facets of the cloth both below and above it in the stacked geometry. As we assume that the friction between two pieces of the cloth is infinite, this vertex will not be able to move as a result of gravity, and need not be grasped. Using the heights stored at each facet, we can compute for each vertex whether it is covered or not.

This defines fully what vertices need to be grasped to achieve a g-fold of stacked geometry. If any such vertex is not on the boundary of the silhouette of the stacked geometry, the g-fold is infeasible (for example, the second g-fold of Fig. 4 (a) is infeasible for this reason). The 3-D motion of the grippers can be computed in the same way as for planar geometry, as discussed in Section 4.1. The running time for computing the vertices that need to be grasped is in principle exponential in the number of g-folds that preceeded, as in the worst case i g-folds create 2^i facets. If we

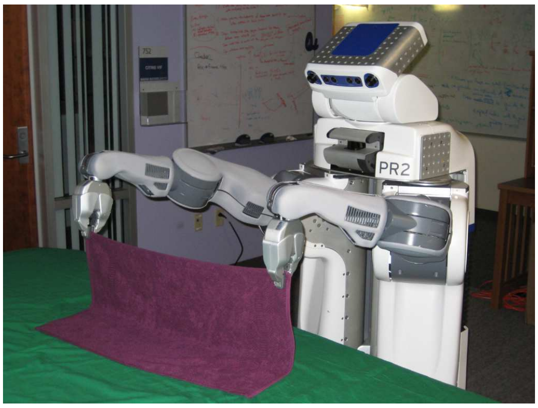


Fig. 8 The PR2 robotic platform (developed by Willow Garage) performing a g-fold on a towel.

consider the number of g-folds a constant, the set of vertices that need to be grasped can be identified in $O(n)$ time.

After the g-fold is executed, we need to update the data fields of the facets that were folded in the geometry: each of their associated transformation matrices is *pre*-multiplied by the matrix F_i corresponding to a mirroring in the line segment specifying the g-fold (F_6 in Fig. 6(b) for the t-shirt example). The stack height of these facets is updated as follows: the order of the heights of all facets that are folded are *reversed* and put on top of the stack. In the example of Fig. 7(a), the facets that are folded have heights 4, 6, 1, and 3 before the g-fold, and heights 8, 7, 10, and 9 after the g-fold, respectively.

The above procedure can be executed in series for a sequence of g-folds. Initially, the geometry has one facet (the original polygon) with height 1 and transformation matrix I (the identity matrix). If a g-fold is specified to only apply to the folded part of the geometry of the last g-fold (a “red” g-fold), the procedure is the same, but only applies to those facets that were folded in the last g-fold. We allow these kinds of g-folds as a special primitive if they need the same set of vertices to be grasped as the previous g-fold. Even if the vertices that are grasped are not on the boundary of the silhouette of the geometry, the g-fold can be achieved by not releasing the vertices after the previous g-fold. This enriches the set of feasible fold primitives.

5 Experiments

5.1 Experimental Setup

We used a Willow Garage PR2 robotic platform [15], shown in Fig. 8. The PR2 has two articulated 7-axis arms with parallel jaw grippers. We used a soft working surface, so the relatively thick grippers can easily get underneath the cloth. Our

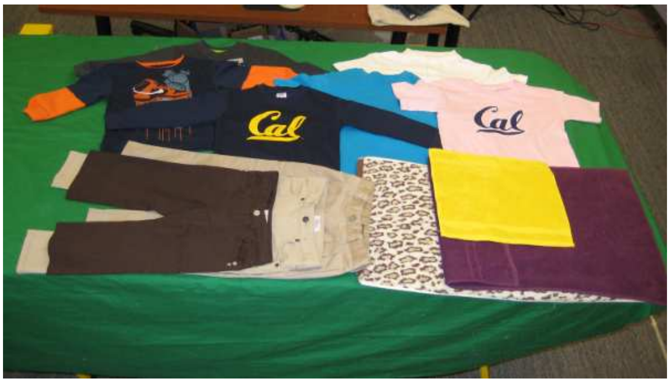


Fig. 9 Three of each clothing category were used in conducting our experiments.

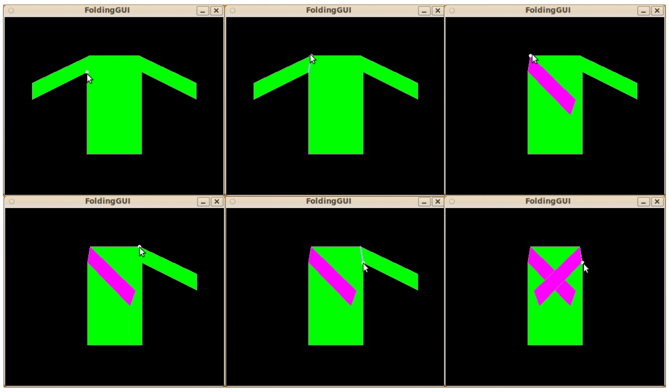


Fig. 10 An example sequence of user-specified folds. The user first clicks on the left arm-pit, then on the left shoulder to specify the first fold. The GUI then verifies that this is a valid g-fold for the chosen number of grippers. In this case it is, and it then shows the result after executing the g-fold (3rd image in the top row). Then the user specifies the next fold by two clicks, the program verifies whether it's a valid g-fold, and then shows the result after executing the g-fold.

approach completely specifies end-effector position trajectories. It also specifies the orientation of the parallel jaw grippers' planes. We used a combination of native IK tools and a simple linear controller to plan the joint trajectories.

We experimented with the clothing articles shown in Fig. 9. Whenever presented with a new, spread-out clothing article, a human user clicks on the vertices of the article in an image. The user is then presented with a GUI that allows them to specify a sequence of folds achievable with the two grippers. Once a valid g-fold has been specified, the GUI executes the fold and the user can enter the next fold. Fig. 10 illustrates the fold sequence specification process through an example.

As many sophisticated folds are difficult, if not impossible, to perfectly define by hand, the GUI is also seeded with a set of folding primitives. When presented with a particular article of clothing, the user is given the option of calling one of

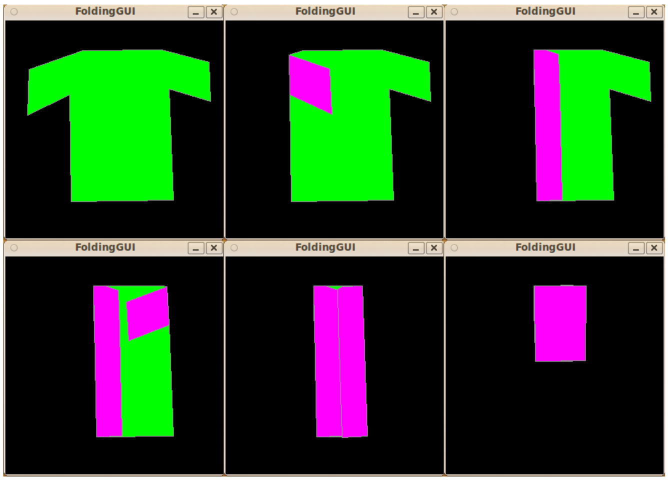


Fig. 11 An example folding primitive, automatically executed on a T-shirt polygon. Note the clean fold, despite the imperfect symmetry of the original polygon.

these primitives. Once called, a sequence of folds is computed, parametrized on a number of features such as scaling, rotation, and side lengths. Fig. 11 shows an example primitive being executed on a user-defined polygon in the shape of a shirt. To ensure consistency across multiple trials, such primitives were used to execute the folds detailed in the Experimental Results section below.

While our approach assumes the cloth has zero resistance against bending, real cloth does indeed resist against bending. As a consequence, our approach outlined so far overestimates the number of grippers required to hold a piece of cloth in a predictable, spread-out configuration. Similarly, our robot grippers have non-zero size, also resulting in an overestimation of the number of grippers required. To account for both of these factors, our implementation offers the option to allocate a radius to each of our grippers, and we consider a point being gripped whenever it falls inside this radius. To compute the grip points, we first compute the grip points required for point grippers and infinitely flexible cloth. We then cluster these points using a simple greedy approach. We begin by attempting to position a circle of fixed radius in the coordinate frame such that it covers the maximum number of grip points, while subsequently minimizing the average distance from each covered point to its center. This process is iterated until no point remains uncovered. For the duration of the fold, our grippers now follow the trajectory of the center of each cluster, rather than individual grip points.

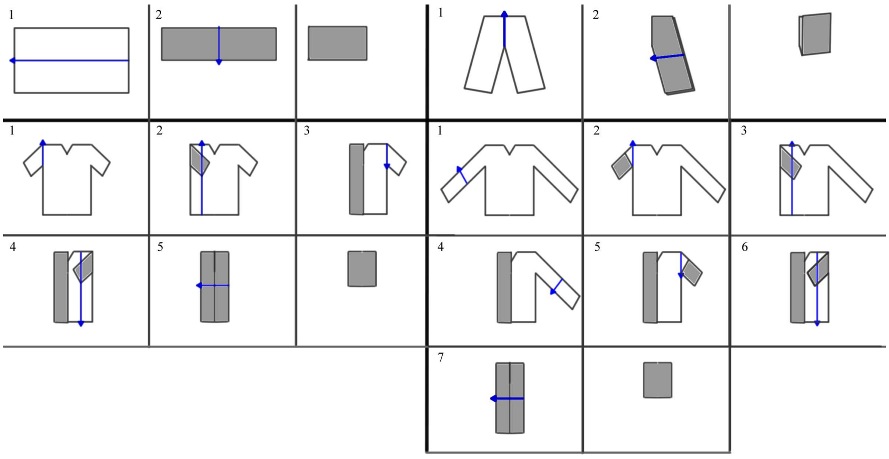


Fig. 12 The user-requested sequences of folds used in our experiments.

Category	Success rate	Avg time (s)	Category	Success rate	Avg time (s)
Towels	**9/9**	**200.0**	**Short-Sleeved Shirts**	**7/9**	**337.6**
Purple	3/3	215.6	Pink T-Shirt	2/3	332.8
Leopard	3/3	210.9	Blue T-Shirt	2/3	343.2
Yellow	3/3	173.5	White Collared	3/3	337.6
Pants	**7/9**	**186.6**	**Long-Sleeved Tops**	**5/9**	**439.0**
Long Khaki	3/3	184.9	Long-Sleeved Shirt	2/3	400.7
Brown	1/3	185.9	Gray Sweater	1/3	458.4
Short Khaki	3/3	189.1	Blue Sweater	2/3	457.8

Table 1 Experimental results of autonomous cloth folding.

5.2 Experimental Results

We tested our approach on four categories: towels, pants, short-sleeved shirts, and sweaters. Fig. 12 shows the fold sequences used for each category. To verify robustness of our approach, we tested on three instances of each category of clothing. These instances varied in size, proportion, thickness, and texture. At the beginning of each experimental trial, we provided the PR2 with the silhouette of the polygon through clicking on the vertices in two stereo images.

Fig. 13 shows the robot going through a sequence of folds. Table 1 shows success rates and timing on all clothing articles. As illustrated by these success rates, our method demonstrates a consistent level of reliability on real cloth, even when the manipulated fabric notably strays from the assumptions of our model. For instance, the g-fold method worked reasonably well on pants, despite the material's clear violation of the assumption of non-zero thickness, and a three-dimensional shape which was not quite polygonal. It was also able to fold a collared shirt with perfect accuracy, despite a rigid collar and buttons, neither of which are expressible in the language of our model. This level of practical success is indicative of a certain ro-

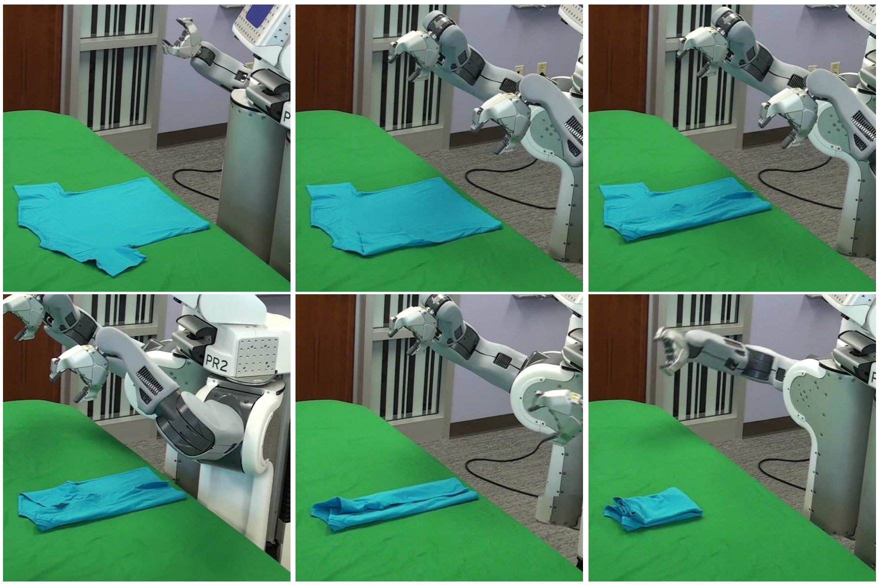


Fig. 13 The robot folding a t-shirt using our approach.

bustness to our approach, which lends itself to a number of implications. The first is that despite the simplifications inherent to our model, real cloth behaves deterministically under roughly the same conditions as its ideal counterpart. While human manipulation of cloth exploits a number of features which our model neglects, these features generally arise in states which our model considers unreachable. That is, handling true fabric often requires less caution than our g-fold model predicts, but rarely does it require more. The second is that even when unpredicted effects do arise, the final result is not compromised. Although factors such as thickness may cause the cloth to deviate slightly from its predicted trajectory – most often in the form of "clumping" for thick fabrics – the resulting fold generally agrees with the model, particularly after smoothing.

Much of our success can be attributed to a number of assumptions which real cloth very closely met: namely, the infinite friction between the cloth and the table, and the infinite friction between the cloth and itself. The former allowed us to execute g-folds even when the modeled polygon did not perfectly match the silhouette of the cloth. As actual articles of clothing are not comprised solely of well-defined corners, this imprecision often resulted in a nonzero horizontal tension in the cloth during the folding procedure. However, as the friction between the cloth and the table far outweighs this tension, the cloth remained static. The latter allowed us to stabilize loose vertices by "sandwiching" them between two gripped portions of cloth. This technique, in combination with the robust gripping approach detailed above, allowed us to execute a number of folds (such as the shirt folds in Fig. 12) which more closely resembled their standard human counterpart. With the exception of long-sleeved shirts, all sequences could be perfectly executed by a pair of point grippers. However, some relied on the ability to create perfect 90 degree angles, or perfectly align two edges which (in actuality) were not entirely straight. Exact precision was impossible in both of these cases; but where there was danger of gravity influencing a slightly unsupported vertex, the friction of the cloth, in conjunction with its stiffness, often kept it in a stable configuration.

The trials were not, however, without error. Most often, failure was due to the limitations of our physical control, rather than a flaw in our model. For instance, 2/2 Short-Sleeved failures and 3/4 Long-Sleeved failure occurred at steps where the robot was required to grasp a portion of previously folded sleeve (Short-Sleeve Steps 2 and 4, Long-Sleeve Steps 3 and 6 in Fig. 12) In all cases, the failure could be easily predicted from the location of the initial grasp. Either the robot did not reach far enough and grasped nothing, or reached too far and heavily bunched the cloth. These failures suggest a clear issue with our implementation: namely, the reliance on open-loop control. While the initial position of each vertex is given, the location of a folded vertex must be derived geometrically. For this location to be correct, we must make two assumptions: that the cloth at hand is perfectly represented by the given polygon, and that the trajectory, once computed, can be exactly followed. Clearly, both are idealizations: the former disregards the multi-layered nature of all but towels (which saw a 100% success rate) and the latter is hindered by the inherent imprecision of any robotic mechanism. A closed-loop method, which would allow the robot to adjust the shape of the modeled polygon to allow for real-world discrepancies, would likely eliminate these issues.

In addition, the robot moved very slowly in our trials, leading to large running times. The reason for moving slowly, besides the fact that it helps the cloth behave in a more deterministic fashion (no dynamics effects), is that it led to higher accuracy of the motions of the arms and base of the robot, which was required for open-loop control. Also here, a closed-loop method would likely enable performance improvements.

Videos of our experimental results are available at http://rll.berkeley.edu/wafr10-gfolds/.

6 Conclusion and Future Work

We described a geometric approach to cloth folding—avoiding the difficulties with physical simulation of cloth. To do so, we restrict attention to a limited subset of cloth configuration space. Our experiments show that (i) this suffices to capture interesting folds, and (ii) real cloth behaves benignly, even when moderately violating our assumptions. Our approach enabled reliable folding of a wide variety of clothing articles.

In the experiments we performed, a human user had to click on the vertices of the clothing article, and would then select a fold sequence. We plan to develop computer vision algorithms that enable automatic recognition of clothing article categories, as well as specific geometric instances thereof. The robot could then look up the desired folding sequence for the presented article. We are also working on simple primitives that will enable taking clothing articles from crumpled, unknown configurations to the spread-out configuration.

Careful inspection of the gripper paths shows that a single very large parallel jaw gripper would suffice to execute a g-fold requiring an arbitrary number of point grippers. We plan to investigate a practical implementation of this idea for the PR2. A large such gripper would reduce the collision free workspace volume significantly.

Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under grants IIS-0905344 and IIS-0904672, by the National Institute of Health (NIH) under grant R01-EB006435, and by Willow Garage by providing the PR-2 robot.

References

1. D. Balkcom, M. Mason. Robotic origami folding. *Int. J. on Robotics Research* 27(5):613–627, 2008.
2. D. Baraff. A. Witkin. Large steps in cloth simulation. *Proc. SIGGRAPH*, 1998.
3. M. Bell. *Flexible object manipulation*. PhD Thesis, Dartmouth College, 2010.
4. M. Bell, D. Balkcom. Grasping non-stretchable cloth polygons. *Int. Journal of Robotics Research*, 29:775–784, 2010.
5. R. Bridson, R. Fedkiw, J. Anderson. Robust treatment of collisions, contact, and friction for cloth animation. *Proc. SIGGRAPH*, 2002.
6. K. Choi, H. Ko. Stable but responsive cloth. *Proc. SIGGRAPH*, 2002.
7. N. Fahantidis, K. Paraschidis, V. Petridis, Z. Doulgeri, L. Petrou, G. Hasapis. Robot handling of flat textile materials. *IEEE Robotics and Automation Magazine* 4(1):34–41, 1997.
8. S. Gupta, D. Bourne, K. Kim, S. Krishnan. Automated process planning for robotic sheet metal bending operations. *Journal of Manufacturing Systems* 17:338–360, 1998.
9. T. Kanade. A theory of origami world. *Artificial Intelligence* 13(3):279–311, 1980.
10. R. Lang. A computational algorithm for origami design. *Proc. Symp. on Computational Geometry*, 1996.
11. J. Liu, J. Dai. An approach to carton-folding trajectory planning using dual robotic fingers. *Robotics and Autonomous Systems* 42:47–63, 2003.
12. J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2010.
13. L. Lu, S. Akella. Folding cartons with fixtures: a motion planning approach. *IEEE Trans. on Robotics and Automation* 16(4):346–356, 2000.
14. F. Osawa, H. Seki, Y. Kamiya. Clothes folding task by tool-using robot. *Journal of Robotics and Mechatronics*, 2006.
15. K. Wyrobek, E. Berger, H.F.M. Van der Loos, K. Salisbury. Towards a Personal Robotics Development Platform: Rationale and Design of an Intrinsically Safe Personal Robot. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008.