Cloth Region Segmentation for Robust Grasp Selection

Jianing Qian*, Thomas Weng*, Luxin Zhang, Brian Okorn, David Held

***Abstract*— Cloth detection and manipulation is a common task in domestic and industrial settings, yet such tasks remain a challenge for robots due to cloth deformability. Furthermore, in many cloth-related tasks like laundry folding and bed making, it is crucial to manipulate specific regions like edges and corners, as opposed to folds. In this work, we focus on the problem of segmenting and grasping these key regions. Our approach trains a network to segment the edges and corners of a cloth from a depth image, distinguishing such regions from wrinkles or folds. We also provide a novel algorithm for estimating the grasp location, direction, and directional uncertainty from the segmentation. We demonstrate our method on a real robot system and show that it outperforms baseline methods on grasping success. Video and other supplementary materials are available at: https://sites.google.com/view/cloth-segmentation.**

I. Introduction

Manipulating and interacting with cloth is a key part of daily life, yet cloth manipulation by robots remains a challenging problem. Cloth is difficult to perceive and manipulate because its deformable nature breaks the rigid-body assumptions of many algorithms. For example, most pose estimation algorithms assume that objects can only transform in 6 degrees of freedom (translation and rotation). However, cloth can deform at any location and thus has nearly an infinite number of degrees of freedom.

In cloth-based tasks like laundry folding and textile manufacturing, it is important to detect and grasp specific regions of cloth, e.g. corners and edges, for downstream manipulation like folding or smoothing. These edges and corners are distinct from wrinkles and folds, which are less useful for downstream tasks.

In order to grasp the cloth along an edge or corner, we must not only detect the cloth edges and corners but also estimate the appropriate grasping direction. Given a grasp position, the grasp direction specifies the approach vector the gripper follows towards this point. Although estimating the grasping direction would be relatively simple if the cloth were lying flat on the table, it is much more challenging in crumpled configurations. Much work has been done for perception and manipulation of cloth in both randomized and predefined cloth configurations, yet cloth-related tasks like laundry folding and assisted dressing remain challenging due to the inherent complexity of cloth dynamics.

In this paper, we present an approach for segmenting these key regions of cloth, even in highly crumpled configurations. To achieve this, we train a neural network to predict cloth edges and corners from a depth image. We also train the network to predict the inner edges, the region interior to the cloth's true edges, for grasp direction estimation. The network is trained on a dataset of RGB-D images extracted from 8 minutes of video of a human manipulating the cloth. The ground-truth for the network is provided by color-labeling the cloth (see Fig. 1), forgoing the need for expensive human annotations.

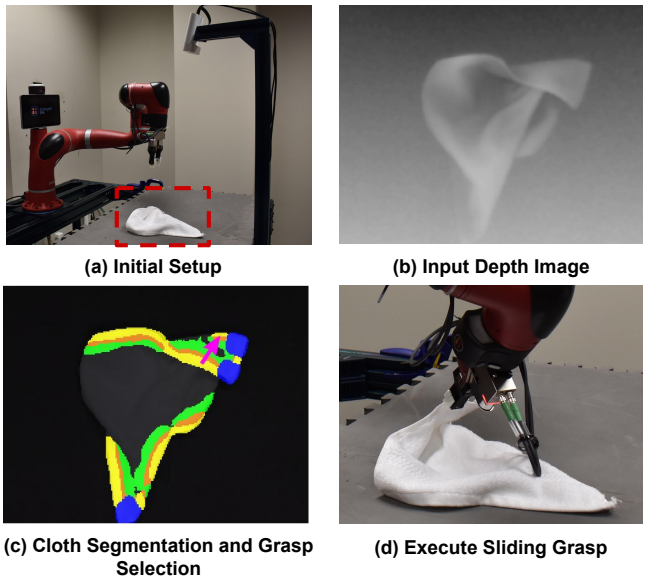


Fig. 1: Grasping using cloth region segmentation: Robot with depth sensor (a) captures depth image of test cloth (b). Depth image is segmented into outer edges (yellow), inner edges (green) and corners (blue) using our cloth region segmentation network (c). Ambiguous regions are colored in orange. Our method selects a grasp location and direction, shown as a magenta arrow. The robot executes a sliding grasp and successfully grips the cloth by its edge.

The segmentation output of our network allows us to quickly and robustly estimate the appropriate position and grasp direction from a crumpled cloth. It also allows us to estimate the grasp directional uncertainty for every edge/corner pixel. This estimation is important for grasping the cloth, as mis-estimating the grasp direction and approaching at an angle not orthogonal to the cloth edge is more likely to fail. Using a dense estimate of grasp directional uncertainty, we can choose the grasp point most likely to succeed.

We implement our method on a real robot system and evaluate its performance on grasp success metrics against a number of baselines; this evaluation demonstrates the

Jianing Qian, Thomas Weng, Luxin Zhang, Brian Okorn, and David Held are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. {jianingq, tweng, luxinz, bokorn, dheld}@andrew.cmu.edu

*These authors contributed equally and are listed in alphabetical order.

strength of our system in estimating cloth edge and corner positions, grasp direction, and grasp uncertainty.

Our contributions include:

- A method to segment regions of cloth critical for downstream manipulation tasks.
- An algorithm to determine a robust grasp configuration accounting for uncertainty about the cloth direction.
- An evaluation of our method against baselines on a real robot system for grasping edges and corners of cloth in crumpled configurations.

II. Related Work

A. Cloth Perception

Robotic cloth manipulation is a well-studied domain with a variety of unsolved tasks, including laundry folding [1], [2], laundry unfolding or smoothing [3], [4], [5], [6], [7], [8], bed making [9], [10], and grasping [11], [12], [13].

Many of these approaches use traditional computer vision algorithms to detect cloth regions for various downstream tasks: [6] chooses candidate grasp points by using Harris corner detection and discontinuity checks on the depth image for peak ridges and peak corners. [3] uses a pre-task manipulation, lifting the towel into the air and shaking it to remove wrinkles before returning it to the table. Canny edge detection is then used to compute contours for interior and exterior corner classification. [1] performs background subtraction and uses stereo images to select a centered point in a pile of towels. They grasp the towel from a central point and and rotate it to obtain a sequence of images. Towel corners are fit to these images using RANSAC. These perception algorithms usually require significant pre-manipulations to get a more structured configuration of the cloth, thus they are more time consuming than many learning-based methods. Furthermore, without these pre-manipulations, these methods are likely to fail under difficult initial configurations, such as highly crumpled cloth. We will show in Sec. IV-B that our method is much more robust to these crumpled cloth configurations compared to these traditional methods.

Another group of methods apply learning-based algorithms for image feature extraction. [11] uses the YOLO detection network to detect the thickest folded edge and grasp a folded towel from a stack. [8] uses an autoencoder network to predict the real edges of towels. This is similar to our approach; however, their method trains a network to output latent features and performs nearest-neighbor classification on input features to predict good grasp points, whereas our network *directly* outputs segmentation masks of grasp regions and also determines good *grasp directions*. Their method also operates on RGB images and requires a human-annotated dataset of corners, whereas our method takes depth images as input to be invariant to changes in visual texture, and does not require human labeling.

The most similar method to ours is [10] which learns to identify a corner of a bed sheet by painting the corner red. Our method expands upon this work by estimating a *dense segmentation* of multiple real edges, inner edges, and corners, as opposed to regressing to a *single* corner position. Furthermore, our method outputs dense grasp direction proposals as well as their corresponding uncertainty estimates. As we will show in Sec. IV-B, the grasp direction proposals and uncertainty estimates are crucial for our performance on our grasping evaluation. Specifically, these two outputs enable us to handle challenging crumpled cloth configurations.

B. Cloth Grasping

Although the focus of our work is on perception rather than grasping, we review prior work on cloth grasping strategies. A simple top-down or angled grasp is commonly used once a grasp point has been selected [6], [10]. A top-down grasp followed by a 6DOF grasping on detected corners of the the hanging cloth has also been studied [1].

Other prior works learn a policy for grasping. [12] learns parameters for motion and grasp primitives to grasp a folded towel. [11] uses Q-learning to train a policy for grasping a folded towel from a stack. [13] uses Soft-Actor-Critic to train a policy for rope and cloth manipulation.

In our work, we identify the real corners and edges of the cloth and select a robust grasping point. Then we execute a hand-designed sliding grasp policy on the selected grasping point in order to pick up the cloth by a single edge or corner.

III. Approach

A. Problem Statement

In cloth manipulation tasks such as laundry folding, it is important that the robot be able to identify and grasp key regions of the cloth. These regions typically include the "real edges" or corners of a cloth. By "real edges," we mean the edges of the cloth in the unfolded configuration, as opposed to any folds or creases that may appear as edges in a particular configuration. If the robot grasps a cloth fold or crease and attempts to use such a grasp to neatly fold the cloth, the result likely will not end up as expected. Thus, failing to grasp the cloth along the real edges could lead to failures for many downstream tasks.

As we will show, traditional computer vision algorithms fail to distinguish the difference between a real cloth edges and apparent edges created by creases or folds. In addition, the robot must also determine the appropriate grasping direction along the cloth edge, which is non-trivial if the cloth is in a crumpled configuration; we will show that simple heuristics frequently fail at this task. In this section, we provide a method that identifies edges and corners of a cloth, predicts grasp directions, and estimates the uncertainty of these directions. These predictions will then be used to quickly and reliably grasp the cloth along its edges and corners, even from crumpled configurations.

B. Method Overview

Fig. 2 provides the overall pipeline of our method. First, our segmentation network takes in a depth image and predicts the outer edges, inner edges and corners. Based on the segmentation, we estimate the grasp direction by computing a correspondence between outer edge and inner edge points.

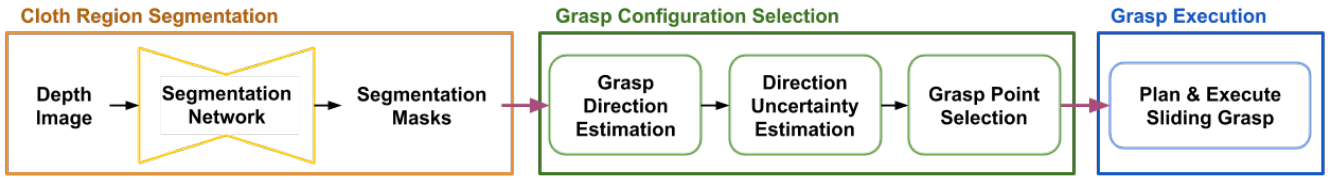


Fig. 2: Pipeline for our method. Cloth region segmentation takes a depth image and outputs segmentation masks for cloth edges and corners. Grasp selection uses the masks to compute a grasp point and direction in the camera frame. Grasp execution transforms the grasp configuration into the robot frame and executes the grasp.

Next, we compute the grasp direction estimation and select a grasp point based on our uncertainty estimate $\mathbf{U}(\mathbf{p})$ for an outer edge point $\mathbf{p}$. Finally, we estimate the 6D robot pre-grasp pose based on the grasp point selected and execute our sliding grasp policy. These components are explained in greater detail in the following sections.

C. Cloth Region Segmentation

We frame the problem of identifying important regions cloth as semantic segmentation. We train a neural network which receives as input a depth image of the scene containing the cloth. The network predicts semantic labels for each pixel, giving the probability that the pixel contains a cloth outer edge, inner edge, corner, or none of these. We can then threshold this probability to obtain a semantic segmentation mask for the cloth edge and corner locations. Fig. 1c shows an example output of our network.

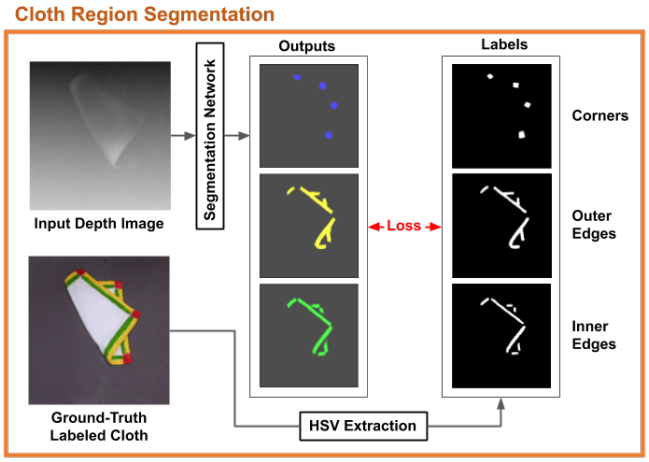


Fig. 3: Training the segmentation network. The network receives a depth image as input. A paired RGB image supervises the network through the color labels of the cloth. Different colors are used to label the corners, outer edges, and inner edges. The ground-truth color for corner labels was changed from red to blue in the outputs to be color-blind friendly.

To train such a network, we need ground-truth labels for the cloth edges and corners. Unfortunately, these are difficult to obtain in images with crumpled cloth, as this would require a large amount of human annotation effort. Instead, we adopt an approach similar to that of [10], in which they mark a single corner of a cloth with a red marker, and train a network to regress to the single corner location. In our case, we mark all edges and corners with different colors of paint and set up the problem as semantic segmentation, to estimate the position of all cloth edges and corners in the image (other differences from [10] are explained in Sec. II).

As we will show, these labels will allow our network to differentiate between real edges or corners of the cloth from cloth folds, which may appear similar to edges in an image. Fig. 3 is a visualization of our training method.

To train the segmentation network parameters θ using these labels, we define the loss $\mathcal{L}$ to be the mean of the pixel-wise binary cross-entropy loss ℓ_k for each class $k \in K$:

$$\mathcal{L}(\theta) = \frac{1}{K} \sum_{k}^{K} \ell_k \tag{1a}$$

$$\ell_k = -\sum_{i \in I} w_k (y_i \log \hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{1b}$$

where i is a pixel in the input depth image I, w_k is a per-class weight to handle the imbalanced distribution between positive and negative labels, y_i is the binary pixel label, and $\hat{y}_i$ is the network prediction for pixel i.

D. Grasp Configuration Selection

1) Grasp Direction Estimation: Once the edges and corners are estimated, the next step is to determine the appropriate grasp direction. To achieve this, we augment the above pipeline by also predicting the cloth "inner edges." We define the cloth outer edge as the region within 1.5 cm of the cloth edge, the cloth corners as the region within 3×3 cm of the corner, and the inner edge as a 1.5 cm region interior to the cloth outer edge. The inner edge labels are shown in green in Fig 3. As before, we obtain cloth inner edge ground-truth labels using another color to paint the inner edge of a cloth, and we train a neural network to predict the cloth inner edge from a depth image.

Given the predicted segmentation for these cloth regions, we now select a grasp point and direction. We want to select the direction that allows our sliding grasp policy to most easily grasp the cloth. A sliding grasp that starts with the gripper oriented towards a cloth edge as in Fig. 5 will intercept the edge upon translation. However, a grasp oriented parallel to the edge or approaching from the reverse direction will not intercept the edge and will fail to grasp.

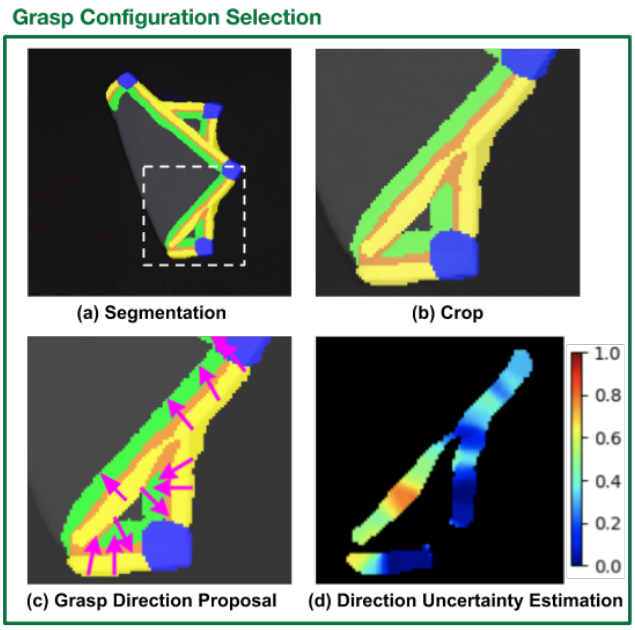


Fig. 4: Illustration of grasp configuration selection. Corners are labeled in blue, outer edges in yellow, inner edges in green. Overlapping outer edge and inner edge segmentations are in orange; After obtaining the cloth region segmentation, (b) shows the cropped section in (a); (c) shows a subsample of grasp direction proposals for each outer edge points; (d) shows the grasp directional uncertainty for each outer edge points.

Grasp direction is similarly important for corners, as sliding grasps that approach the corner head-on or aligned with the edge of the cloth are more likely to succeed than other orientations.

The following is our procedure for computing the appropriate grasp direction. We first threshold the output of the network described in Sec. III-C to obtain a set of points estimated to belong to the outer edge $\mathbf{E_O}$ and a set of points that belong to the inner edge $\mathbf{E_I}$. Then, for each outer edge point $\mathbf{p} = [p_x, p_y] \in \mathbf{E_O}$, we find the closest inner edge point $\mathbf{q}^* = [q_x, q_y]$. More formally, we define $\mathbf{q}^*$ to be

$$\mathbf{q}^* = \underset{\mathbf{q} \in \mathbf{E_I}}{\arg\min} \|\mathbf{p} - \mathbf{q}\|_2 \tag{2}$$

With the correspondence between $\mathbf{p}$ and $\mathbf{q}^*$, we further define the grasp direction at point $\mathbf{p}$ to be the direction along the vector from $\mathbf{p}$ to $\mathbf{q}^*$. Fig. 4c shows a subset of those grasp directions. The vector from $\mathbf{p}$ to $\mathbf{q}^*$ often defines an appropriate grasp direction at point $\mathbf{p}$. This direction can be used by the robot to grasp the cloth.

2) Directional Uncertainty Estimation: Fig. 4c also shows a few cases where, due to the complex folds of the cloth, the vector from $\mathbf{p}$ to $\mathbf{q}^*$ does not indicate an appropriate grasp direction. Thus, for robust grasping, we also compute a measure of the uncertainty in this grasp direction.

We define the uncertainty of the grasp direction for a single point $\mathbf{p}$ to be the variance of the grasp directions predicted by its neighbours. To compute this variance, let $\mathbf{N_k}(\mathbf{p})$ be the set of k closest pixel points in $\mathbf{E_O}$ of $\mathbf{p}$ in Euclidean distance; let α be the angle between $\overrightarrow{\mathbf{pq}^*}$ and a unit vector along the horizontal x axis. Formally we can define the cosine and sine of the grasp direction at $\mathbf{p}$ as

$$f_{cos}(\mathbf{p}) = cos(\alpha) = \frac{q_x - p_x}{\|\mathbf{q}^* - \mathbf{p}\|_2} \tag{3}$$

$$f_{sin}(\mathbf{p}) = sin(\alpha) = \frac{q_y - p_y}{\|\mathbf{q}^* - \mathbf{p}\|_2} \tag{4}$$

We can then define observation vectors $\mathbf{x_0}(\mathbf{p})$ and $\mathbf{x_1}(\mathbf{p})$ to contain the cosine and sine of the grasp direction of all points in $\mathbf{N_k}(\mathbf{p})$:

$$\mathbf{x_0}(\mathbf{p}) = \left\{ f_{cos}(n) \mid n \in \mathbf{N_k}(\mathbf{p}) \right\} \tag{5}$$

$$\mathbf{x_1}(\mathbf{p}) = \left\{ f_{sin}(n) \mid n \in \mathbf{N_k}(\mathbf{p}) \right\} \tag{6}$$

Next, we define the sample covariance matrix $\mathbf{K}(\mathbf{p})$ in the usual manner from the observations $\mathbf{x_0}(\mathbf{p})$ and $\mathbf{x_1}(\mathbf{p})$

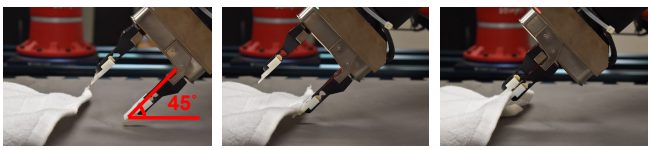
$$\mathbf{K}_{ij}(p) = \frac{1}{N-1} \sum_{k=1}^{N} \left(x_{ik}(p) - \bar{x}_i(p)\right) \left(x_{jk}(p) - \bar{x}_j(p)\right) \tag{7}$$

where $x_{ij}(p)$ is the jth element of $\mathbf{x_i}(\mathbf{p})$, and $\bar{x}_i(p)$ is the mean of $\mathbf{x_i}(\mathbf{p})$.

Finally, we define the uncertainty of our grasp direction prediction to be the sum of the variances of the individual dimensions, or the trace of $\mathbf{K}$: $Tr(\mathbf{K}(\mathbf{p})) = Var(\mathbf{x_0}(\mathbf{p})) + Var(\mathbf{x_1}(\mathbf{p}))$, where $Var(\mathbf{x_i}(\mathbf{p}))$ is the variance of $\mathbf{x_i}(\mathbf{p})$. Since the trace of a matrix is equal to the sum of its eigenvalues, this means that $Tr(\mathbf{K})$ measures the summation of the uncertainty in the principal directions for the covariance matrix $\mathbf{K}$. The trace therefore captures the uncertainty of the grasp direction while being invariant to axis transformations. Fig. 4d shows an example of our uncertainty estimate.

3) Grasp Point Selection: Finally, we describe our method for grasp point selection, which considers the outer edge predictions of Sec. III-C and the directional uncertainty estimates of Sec. III-D.2. For each outer edge point $\mathbf{p} \in \mathbf{E_O}$, we compute an uncertainty estimate $\mathbf{U}(\mathbf{p}) = Tr(\mathbf{K}(\mathbf{p}))$ as described above. Finally, for grasp point selection, we pick the outer edge point $\mathbf{p}$ that has the lowest uncertainty:

$$\mathbf{p} = \underset{\mathbf{p} \in \mathbf{E_O}}{\arg\min}\ \mathbf{U}(\mathbf{p}) \tag{8}$$



(a) Pre-slide pose. (b) Post-slide pose. (c) Pinch grasp.

Fig. 5: Sequence of poses for the sliding grasp policy. The sliding action is a translation from the pre-slide to post-slide pose. The slide intercepts the target grasp point on the cloth.

E. Grasp Execution

Once a grasp configuration with point and direction is chosen, we execute a hand-designed grasping policy to slide one of the gripper's fingertips under the cloth for a pinch grasp. We use this sliding grasp policy instead of a simpler top-down grasping routine, because top-down pinch grasps on edges and corners that are folded over (and hence overlap parts of the cloth) often result in grasping multiple layers of the cloth. A tilted sliding grasp can separate one layer of cloth from another.

The configuration ($\mathbf{p}$, α) specifies the grasp point on the cloth and the direction for the sliding grasp. This configuration is specified in image coordinates; to transform it into the world frame, we perform a 2D-to-3D projection using known camera intrinsics and extrinsics. This provides an intermediate 6D pre-grasp pose $\tilde{\mathbf{g}}$ consisting of the 3D position of the target cloth point (corresponding to $\mathbf{p}$ in 2D), and the 3D orientation of the end-effector (corresponding to α in 2D). The intermediate pre-grasp pose $\tilde{\mathbf{g}}$ is oriented top-down and rotated about the z-axis in the world frame. We apply a final transformation that tilts the grasp pose about the horizontal x-axis by 45-degrees to obtain a new pre-grasp pose $\mathbf{g}$. This pose allows one of the fingertips to get under the cloth during the slide action. This transformation also includes a z-offset to account for the z-height of the gripper tip lowering due to the rotation. Finally, we compute offsets to $\mathbf{g}$ in the xy plane parallel to the workspace to get pre-slide and post-slide poses. As shown in Fig. 5, the sliding grasp policy moves to the pre-slide pose, translates to the post-slide pose, then pinches to grasp the cloth.

F. Implementation Details

1) Network Implementation Details: To train our segmentation network, we collected a dataset of paired RGB-D images. The images were extracted from RGB-D video of a human manipulating a cloth with regions of interest labeled using acrylic paint. The cloth was square, 12 inches each side, and painted with red 3×3 cm corners, yellow 1.5 cm thick outer edges, and green 1.5 cm thick inner edges. See Fig. 3 for an image of the labeled cloth.

The human manipulated this semantically labeled cloth in the robot's workspace by folding it, dropping it, bunching it up, etc. We collected 8 minutes of video for a total of about 6700 RGB-D images. These images were split into 4:1:1 train, validation, and test sets.

Our segmentation network is based on U-Net [14]. We augmented the data during training with random image flips and rotations to improve robustness. Additional details on training and the network architecture are provided in the appendix. All training was performed on an Ubuntu 16.04 machine with an NVIDIA GTX 1080 Ti GPU, a 2.1 GHz Intel Xeon CPU, and 32 GB RAM.

2) Physical Implementation Details: All experiments were performed on a 7 DOF Rethink Robotics Sawyer Robot with a Weiss WSG-32 parallel-jaw gripper. The robot's workspace was a 0.6×0.6 m area. A Microsoft Azure Kinect sensor was mounted 0.7 m above the workspace to provide RGB-D images. Our test cloth is a white, unlabeled cloth with the same dimensions as the labeled one used for training. See Fig. 1a for the complete workspace setup. The default fingertips of the Weiss gripper were too thick to get under the cloth during the sliding maneuver, so we 3D-printed and attached thinner fingertips (see Fig. 5).

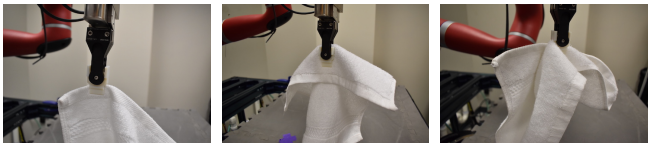
IV. Experiments

Our experiments are designed to answer the following questions:

- How does our learned method for finding cloth edges and corners compare to non-learned baselines?
- How does our method for estimating the grasp direction compare to non-learned baselines?
- Do we obtain more robust grasps using our method for estimating grasp directional uncertainty?

A. Experimental Design

We designed two experiments to evaluate our method against various baselines. The first experiment evaluated performance for grasping cloth edges (as opposed to creases or folds), and the second evaluated grasping cloth corners. In both experiments, each grasping trial starts with a randomly crumpled cloth in the center of the robot's workspace. To enable reproducibility of our results, we used the following protocol in all of our experiments to generate the initial cloth configuration for each trial: at the beginning of each trial, a human grasps the square cloth at the midpoint of an edge. They then hold the cloth at a height such that the lowest point of the cloth is 0.1 m from the workspace surface. Finally, they let go of the cloth from this height to obtain a randomly crumpled cloth pose. This initialization procedure is based on the protocol from [15], adapted for our cloth grasping task.



(a) No fold. (b) Single fold. (c) Multiple folds.

Fig. 6: Examples of cloth grasps. Folds longer than 2cm from edge to fold are considered grasp failures; of these three, only (a) is considered a success.

We define success metrics for grasping the cloth at edges and corners. A grasp is considered a success if it pinches a cloth edge or corner and lifts it 30 cm above the workspace. The flexible and deformable nature of cloth can cause pinch grasps on edges and corners to fold over some of the material. Fig. 6 shows examples of grasps with flat and folded cloth. For grasping edges, we consider a grasp with cloth folded over to be a success if the fold is less than or equal to 2 cm at its maximum length. For grasping corners, we use a threshold of 5 cm from the corner to the fold. These thresholds apply when there is a single cloth fold pinched; if multiple folds are held within the pinch grasp, the grasp is considered a failure.

B. Experimental Results

We evaluate whether our learned method performs better than baselines for identifying cloth edges and corners (as opposed to wrinkles and folds). Our method consists of the cloth region segmentation network, grasp direction estimation, grasp directional uncertainty estimation, and grasp selection, as described in Sec. III.

1) Grasping Cloth Edges: For the task of identifying cloth edges, we evaluate against three baselines:

- "Segment-Edge" segments the cloth from the table using RANSAC plane fitting. A grasp point is randomly selected from the edge pixels of the segmentation. The grasp direction is determined by the direction of the depth gradient at the selected grasp point.
- "Canny-Depth" applies Canny edge detection [16] to the depth image. The grasp point is sampled uniformly from the set of edge points above an intensity threshold. The grasp direction is determined by the depth gradient direction, as in the above.
- "Canny-Color" is the same as Canny-Depth, except it applies Canny edge detection to the gray-scaled color image. The grasp direction is determined by the color gradient direction instead of depth.

See Fig. 7 for visualizations of these methods.

The results are shown in Table I. We performed 3 trials with 10 grasps each to estimate a mean and variance for each method. Our method significantly outperforms the baselines in terms of grasp success. The network is largely able to correctly distinguish between edges and folds, determine an appropriate grasp configuration direction, and execute a successful grasp. Averaging over the trials, there were an average of 2.7 failures out of 10 grasps due to misdetection, meaning that the grasp point selected was not a real edge. There was an average of 0.3 failures out of 10 grasps due to failed grasping. See Sec. IV-B.4 for more details on failure cases.

The baselines perform poorly largely due to an inability to distinguish between real cloth edges versus folds. Canny-Depth relies on the intensity of depth gradients to find cloth edges, but depth gradients occur for both cloth edges and large folds. Segment-Edge fails due to noisy segmentation; because the cloth is thin, parts of the cloth can fall within the inlier threshold of the RANSAC table segmentation, despite careful parameter tuning. Still, even with a clean segmentation, grasping at an edge point on the segmentation mask often results in grasping a cloth fold for our highly crumpled cloth configurations. Canny-Color uses color gradients to find edges. It is less affected by noise compared to the depth-based baselines, as the white cloth stands out from the darker background of the table, resulting in strong edges. However, this method is still unable to discriminate between real cloth edges from folds, resulting in failure in a majority of grasp attempts.

Our network is able to perform better than all of these baselines by using a learned segmentation. The successful grasps are also of higher quality, meaning that the grasps are more often flat with no folding of the cloth, and the edge is horizontal to the gripper tip. In terms of execution time, the perception component of our method runs in approximately 0.25s, with the segmentation network contributing approximately 0.14s to that total. Grasp execution is a larger bottleneck and requires approximately 15s for all methods.

TABLE I: Grasping Cloth Edges

Method	Grasp Success
Canny-Depth	0.20 ± 0.00
Segment-Edge	0.30 ± 0.00
Canny-Color	0.33 ± 0.12
Our Method	$\mathbf{0.70 \pm 0.20}$

3 trials per method, 10 grasp attempts per trial

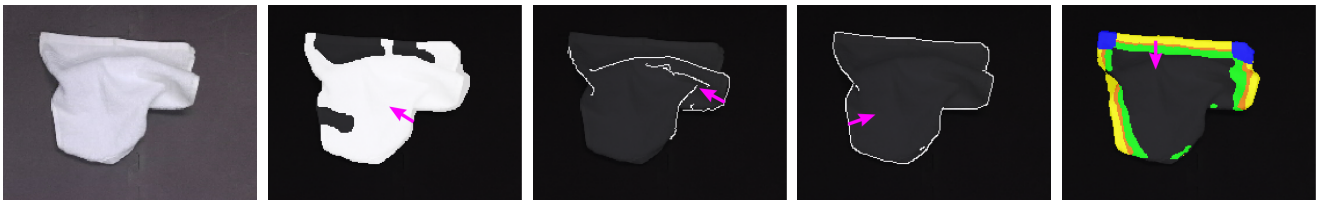
2) Grasping Cloth Corners: We also evaluated our method on grasping corners. Our method remains the same, except that corners are used for grasp point selection instead of edges. The corners still use correspondence with inner edges to determine grasp direction, as well as our method for estimating grasp directional uncertainty described in Sec. III.

For this task, we evaluated against the following baselines:

- "Harris-Depth" applies Harris corner detection [17] to the depth image. The maximum intensity value is selected as the grasp point. The depth gradient direction at the grasp point is used to determine the grasping direction, as in the edge grasping experiments.
- "Harris-Color" takes a grayscaled RGB image as input and uses color gradients to determine the grasping direction, but is otherwise the same as the above.

The results are shown in Table II. Our method outperforms the baselines on corner grasping, being able to more reliably detect corners in any cloth configuration. Averaging over the trials, there were an average of 3 failures out of 10 grasps due to misdetection. There were an average of 1.3 failures out of 10 grasps due to grasping error. Our method performs worse on corners than on edges. Fewer regions of the image are corners compared to edges, so false positives are more problematic. Sec. IV-B.4 for details on failure cases.

The baselines perform poorly for largely the same reason of misdetection as with the edge experiments. The Harris-Depth baseline performs poorly because it looks for large changes in the gradient in all directions, which could result in false positives instead of real corners. Most of the grasp point selections from this baseline were on wrinkles and folds than on the cloth. The Harris-Color baseline performs better than depth, possibly because there are fewer false positives given the white on black input images. White cloth corners against the darker workspace surface can be easily detected; however, corners lying on top of the cloth are less likely to be detected. For our difficult randomly crumpled cloth configurations, the corners are not always cleanly visible against the surface, and often lie in configurations that are difficult to discriminate in 2D.



(a) Cloth Pose (for reference). (b) Segment-Edge. (c) Canny-Depth [16]. (d) Canny-Color [16]. (e) Our Method.

Fig. 7: Segmentation and selected grasp point for edge grasping methods. (b)-(e) visualize the output of each method on top of the reference image (a). Note that the color image is only provided as input to Canny-Color (d); all other methods take the corresponding depth image as input. As shown in (e), our method correctly identifies most of the apparent edges of the cloth as folds, whereas the other methods fail to make this distinction.

TABLE II: Grasping Cloth Corners

Method	Grasp Success
Harris-Depth	0.05 ± 0.07
Harris-Color	0.33 ± 0.15
Our Method	$\mathbf{0.57 \pm 0.06}$

3 trials per method, 10 grasp attempts per trial

3) Ablations: We perform ablations on our method to determine the relative contribution of the different components of our method to grasp success. Our full method consists of segmenting cloth regions using a neural network (Sec. III-C), determining the grasp direction for all segmented edge/corner pixels using their nearest segmented inner edge pixels (Sec. III-D.1), and selecting a grasp point with the lowest grasp directional uncertainty (Sec. III-D.2).

We perform the following ablations of our method:

- "No-Direction-Prediction" still uses the cloth segmentation network of Sec. III-C. However, rather than determining the grasp direction using the methods of Sec. III-D.1 and Sec. III-D.2, this ablation determines the grasp direction by fitting a bounding box around the segmented outer edge pixels and setting the direction to be the vector pointing to the center of the box. Instead of using the point with minimum directional uncertainty, it randomly selects the grasp point from the set of outer edge pixels.
- "No-Directional-Uncertainty" still uses the cloth segmentation network of Sec. III-C as well as the method of Sec. III-D.1 for determining the grasp direction. However, rather than computing the grasp directional uncertainty to choose a grasp point as in Sec. III-D.2, this ablation chooses a grasp point randomly.

The results are shown in Table III. The ablations underperform the full method, demonstrating that our method for estimating the grasp direction (Sec. III-D.1) as well as our method for estimating directional uncertainty (Sec. III-D.2) help to choose more robust grasps. We observe No-Direction-Prediction selecting grasp directions near-parallel to real edges instead of orthogonally, because it always chooses directions toward the center of the segmentation bounding box. The performance of No-Directional-Uncertainty vs. No-Direction-Prediction provides evidence that using the inner edge segmentation to determine the grasp direction improves grasp success. Comparing our full method with No-Directional-Uncertainty shows that selecting the grasp point with minimal directional uncertainty outperforms random grasp point selection.

TABLE III: Ablations on Grasping Cloth Edges

Method	Grasp Success
No-Direction-Prediction	0.2
No-Directional-Uncertainty	0.4
Our Method	$\mathbf{0.7 \pm 0.20}$

1 trial per ablation, 10 grasp attempts in trial

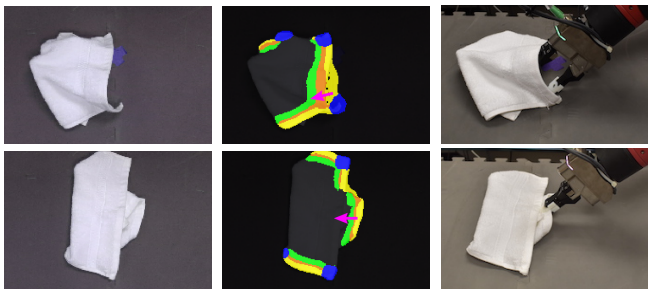
4) Failure Cases: In this section we discuss the most frequent and notable failure cases. Examples of these cases are in Fig. 8 and the supplementary video.

Failures occurred when the segmentation produced by our method contained errors. Because the cloth is very thin and the depth images captured from our sensor are noisy, the network can fail to get accurate segmentation at cloth edges (see Fig. 8, top row). This issue causes both false positives, in which pixels close to real edges are included in the segmentation, and false negatives, in which the segmentation does not include valid pixels. These segmentation errors affect the grasp selection component that takes the segmentation as input. As a result, we sometimes observed our method selecting grasp points on false positives, which were more likely to result in grasp failures.

Failures also occurred due to grasping areas with valid edges but problematic nearby cloth configurations. For example, overlapping edges can create the appearance of a continuous segmentation, and a grasp on that area will result in grasping both edges (see Fig. 8, bottom row). Developing a policy that can adapt to such challenging configurations is an area of future work.

Failures due to motion planning to reach commanded poses happened infrequently, such as when a selected grasp

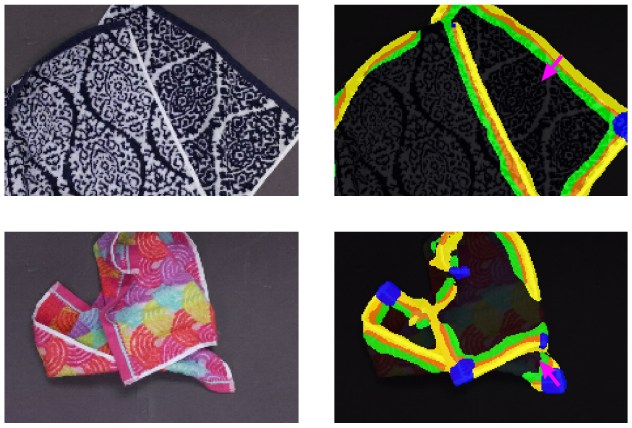
is in an unreachable robot configuration. These failures are easily detected, so we re-execute our method to choose a different grasp point in such cases.



(a) RGB Image. (b) Segmentation and Grasp Prediction. (c) Grasp execution.

Fig. 8: Failure cases. (top row) Segmentation bleeds over real cloth edge, leading to poor estimation of grasp height. (bottom row) Grasp fails to avoid grasping nearby folds and edges (note that misdetection has also occurred).

5) Robustness: We demonstrate that our network is robust to variations in visual texture and cloth size by grasping other cloths (see Fig. 9 and supplementary video). Our network can segment cloth with different colors and patterns because it only takes depth as input. It can also segment cloths of different dimensions due to its fully convolutional architecture.



(a) RGB Image. (b) Segmentation and Selected Grasp.

Fig. 9: Our network is able to segment cloths of various sizes and visual texture. See the supplementary video for grasping demonstrations on these cloths.

V. Conclusion

We present a method to segment real edges and corners of cloth (as opposed to creases or folds) from depth images. Our method also determines a grasp configuration from these segmentations that accounts for directional uncertainty. We demonstrate a system that implements our approach to grasp cloths in crumpled configurations, and we show that our method outperforms various baselines in terms of grasp success rate on grasping success.

Acknowledgment

This work was supported by the National Science Foundation Smart and Autonomous Systems Program (IIS-1849154), the United States Air Force and DARPA under Contract No. FA8750-18-C-0092, LG Electronics, a NSF Graduate Research Fellowship (DGE-1745016), and a NASA Space Technology Research Fellowship (80NSSC17K0233).

References

[1] 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 *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2308–2315.

[2] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 1413–1419.

[3] D. Triantafyllou and N. A. Aspragathos, "A vision system for the unfolding of highly non-rigid objects on a table by one manipulator," in *Intelligent Robotics and Applications*, S. Jeschke, H. Liu, and D. Schilberg, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 509–519.

[4] K. Hamajima and M. Kakikura, "Planning strategy for task of unfolding clothes," *Robotics Auton. Syst.*, vol. 32, pp. 145–152, 1997.

[5] D. Triantafyllou, I. Mariolis, A. Kargakos, S. Malassiotis, and N. A. Aspragathos, "A geometric approach to robotic unfolding of garments," *Robotics Auton. Syst.*, vol. 75, pp. 233–243, 2016.

[6] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 4871–4876.

[7] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 987–993, 2014.

[8] K. Yamazaki, "Gripping positions selection for unfolding a rectangular cloth product," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Aug 2018, pp. 606–611.

[9] M. Laskey, C. Powers, R. Joshi, A. Poursohi, and K. Y. Goldberg, "Learning robust bed making using deep imitation learning with dart," *ArXiv*, vol. abs/1711.02525, 2017.

[10] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep Transfer Learning of Pick Points on Fabric for Robot Bed-Making," in *International Symposium on Robotics Research (ISRR)*, 2019.

[11] S. Demura, K. Sano, W. Nakajima, K. Nagahama, K. Takeshita, and K. Yamazaki, "Picking up one of the folded and stacked towels by a single arm robot," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 1551–1556.

[12] Y. Moriya, D. Tanaka, K. Yamazaki, and K. Takeshita, "A method of picking up a folded fabric product by a single-armed robot," *ROBOMECH Journal*, vol. 5, pp. 1–12, 2018.

[13] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," *ArXiv*, vol. abs/1910.13439, 2019.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[15] I. Garcia-Camacho, M. Lippi, M. C. Welle, H. Yin, R. Antonova, A. Varava, J. Borras, C. Torras, A. Marino, G. Alenyà, *et al.*, "Benchmarking bimanual cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1111–1118, 2020.

[16] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[17] C. G. Harris, M. Stephens, *et al.*, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

APPENDIX

A. Network Architecture

Our network architecture is based on U-Net [14]. It consists of a downsampling part and an upsampling part. In the downsampling path, a step consists of two 3x3 unpadded convolutions, each with batch normalization and a rectified linear unit, followed by a 2x2 max pooling layer with stride 2. We apply four of these steps, doubling the number of feature channels each time. For the upsampling path, a step consists of a 2x2 up-convolution that halves the number of feature channels, a concatenation with a cropped feature map from the corresponding downsampled path, and two 3x3 convolutions, each followed by batch normalization and ReLU. A final 1x1 convolution is used to turn the feature map into 3 classes for corners, outer edges, and inner edges respectively.

The differences between our network and U-Net are that we add batch normalization, and our network takes a single channel depth image as input.

B. Network Training

We implemented the network in PyTorch. We use the Adam optimizer with a learning rate of 1e-5. We use a batch size of 8. To augment our data, we flip the image with 50 percent chance, and also rotate the image with 50 percent chance, sampling within [-30 degrees, 30 degrees].

In our loss function, we set the per-class (corners, outer edges, and corners) weight w_k for balancing the loss on positive and negative predictions to 20 for all classes.

C. Grasp Direction Uncertainty Estimation

As described in Sec. III-D.2, the uncertainty of the grasp direction for a single outer edge point $\mathbf{p}$ is the variance of the grasp directions predicted by its neighbors. Each neighbor is an outer edge pixel with its own grasp direction vector, computed as described in Sec. III-D.1. We form the neighborhood by taking the k outer edge pixel points closest to $\mathbf{p}$, and set $k = 100$.