FlingWAM: Solving Cloth Unfolding Tasks with Angle Value Maps

Adam Kan[1*], Jimmy Wu[2], Jeannette Bohg[3]

[1*]The Nueva School, San Mateo, CA, USA.
[2]Princeton University, Princeton, NJ, USA.
[3]Stanford University, Stanford, CA, USA.

*Corresponding author(s). E-mail(s): adakan@nuevaschool.org;
Contributing authors: jw60@cs.princeton.edu; bohg@stanford.edu;

Abstract

Cloth unfolding tasks are an integral part of household robot operations. For cloth unfolding in common household tasks like organizing laundry, setting tables, and making beds, the goal should always be to achieve maximal cloth coverage. The most effective method of cloth unfolding is currently with dual-arm fling methods. The most pressing research question is currently how to select grasp points for fling actions. We approach this task by utilizing current image segmentation abilities to detect the edges and centroids of cloths. Our method represents grasp points by using angle value maps instead of spatial value maps to take advantage of generalizable physical characteristics of cloth unfolding tasks. We demonstrate the effectiveness of this method in simulation in contrast to other grasp selection methods for dual-arm fling tasks. Our method succeeds in fully unfolding cloths in 23.5% of scenarios, a roughly 80% improvement on the leading alternative method.

Keywords: dynamic manipulation, cloth manipulation, self-supervised learning

1 Introduction

The use of robots to complete common household tasks has been the interest of robotics researchers for years. One step of this far-reaching goal is to build effective and robust programs to complete high-level tasks. Cloth unfolding, from an initial ”messy” state, is one such task with applications in household cleanup and laundry. In this paper, we demonstrate an effective method for solving the task of cloth unfolding, adapted from FlingBot (Ha and Song, 2021).

Our work builds off of two insights from FlingBot. The first is that a dual-arm grasp and fling is the most effective policy for severely self-occluded cloth unfolding. The second is that the primary challenge in the improvement of cloth unfolding models is the selection of grasp points. Given ideal grasp points and an effective fling primitive, a robot can unfold a severely crumpled cloth in just a few interactions.

Cloth unfolding tasks should only be considered successful if a cloth is fully unfolded, i.e. all corners are visible and there are no significant wrinkles. Only at this level of completion is a piece of clothing then able to be neatly folded, or in the case of tablecloths, neatly cover the surface it is designed to cover. Success is achieved in a cloth unfolding task when the area of an operated-on cloth is equal to its flattened area. Cloth unfolding models should be evaluated on their ”success percentage”, or the percent of cloths which achieve complete coverage after having been manipulated by a robot.

FlingWAM utilizes new innovations in open-vocabulary image segmentation (OVIS) to aid a grasp point selection model. Our insights are twofold: 1. Current OVIS models' high accuracy at categorizing and generating masks of cloths can be used to enhance cloth unfolding models, 2. The ideal grasp points for cloth unfolding will always exist on the edges of a cloth.

FlingWAM represents grasp points around the edges of a cloth based on their angle from the centroid of a cloth. Researchers have previously labelled edge grasp points based on their angle from the centroid of a cloth (Willimon and Birchfield, 2011), but haven't proposed methods to directly regress to grasp points defined by angles. Our model generates an angle value map, giving a predicted score for each of 360 angles around the edge of a cloth. The score is regressed to the delta-coverage, or the change in cloth coverage between post-action and pre-action measurements, a metric used by FlingBot.

We demonstrate FlingWAM's success in a simulated environment, with a proposal to implement it in real-world settings with two Kinova robotic arms on motorized bases. Our contributions are: 1. A new measure for success in cloth unfolding tasks, 2. An alternative method for selecting grasp points on a cloth.

2 Related Work

Cloth Unfolding. The task of cloth unfolding has been studied extensively, with most previous research utilizing a single-arm pick-and-drag primitive. While pick-and-drag interactions are effective at resolving small wrinkles in a cloth, they fail when a cloth is severely self-occluded, such as when a corner is tucked under. Pick-and-drag policies require numerous highly precise interactions to unfold severely self-occluded cloths. While pick-and-drag primitives aren't effective, research regarding the selection of grasp points for pick-and-drag primitives can be applied to other primitives.

Researchers have proposed several methods to select grasp points. Some utilize feature detection methods that make use of cloth physics to deconstruct unfolding tasks. These methods isolate corners, ridges, or hemline edges to determine grasp points (Willimon and Birchfield, 2011; Hamajima and Kakikura, 2000). These methods, while effective on partially folded cloths, are ineffective for severely crumpled cloths because corners and hemlines may not be visible and ridges overly prevalent. Another method uses human-labelled datasets which are able to use hemline predictions to predict grasp points (Yamazaki, 2018). This method doesn't rely on algorithmic solutions and is therefore more generalizable, but suffers from its reliance on a small-scale dataset of only 200 scenarios. Successful cloth unfolding policies must be simultaneously generalizable in structure and data requirements.

FlingBot. FlingBot attempts to eliminate the need for specific feature detection or data labelling. The method is a self-supervised training loop that produces spatial value maps directly from images of cloth. Spatial value maps' use in FlingBot is justified by their use in other interactive perception tasks (Wu et al., 2020). FlingBot generates ground truth data by simulating selected grasp points during the training loop.

FlingBot demonstrates high average final coverage on a variety of cloth unfolding tasks when compared to other methods, but several aspects of its structure prevent it from achieving higher final coverage, and especially a high success percentage.

FlingBot ends a cloth interaction whenever the model predicts grasp points not on the cloth. When presented with a near-flattened cloth, FlingBot frequently chooses to end the interaction rather than risk a flinging mistake that decreases cloth coverage. This termination function is effective at stopping the robot before unproductive actions are taken, but prevents the robot from improving on near-perfect scenarios.

FlingBot also suffers from its discretization of grasp angles and diameters. Limited to only 12 angles and 8 diameters, FlingBot is frequently unable to predict the most optimal grasp points, resulting in cloths with corners or edges tucked underneath. A successful grasp selection model must best attempt to imitate the continuous space of grasp point possibilities. FlingBot's methodology requires 96 (12*8) rotated and scaled copies of the workspace image to be inputted to the neural network. To increase the number of predictable grasp angles to 48 would require a 4x increase in model size, significantly slowing training and inference time and memory usage.

Image Segmentation. The most accurate image segmentation models are currently closed-vocabulary, and are exceptionally good at generating masks of seen object categories thanks to significant training datasets (Wang et al., 2023a,b). However, current large-scale datasets don't include cloth or cloth-like object classes, so these models can't be used to generate masks for this task.

Open-vocabulary image segmentation models have also improved significantly in recent years. There are currently several models able to generalize to unseen object types (Gu et al., 2022; Minderer et al., 2022; Kamath et al., 2021).

3 Method

We use the segmentation abilities of a recently developed Mask R-CNN to aid in the detection of a cloth's edges and centroid. We provide this information, along with a top-down view of a cloth, to a CNN modified with residual blocks. Our model generates an angle value map which is converted into two grasp points to execute a fling primitive. We utilize FlingBot's lift, stretch, and fling primitive.

3.1 Image Segmentation

We hypothesize that the most effective places to grasp a cloth for flinging will always exist around the edges of a cloth. To isolate the edges of a cloth, we use an off-the-shelf OVIS model to perform open-vocabulary image segmentation. For our pipeline, we require an accurate mask of a monochrome cloth placed on a gray background.

There are several models capable of achieving high accuracy on our specific task. We select Open-vocabulary Object Detection via Vision and Language Knowledge Distillation (ViLD) (Gu et al., 2022). ViLD outperforms alternative models on the COCO dataset and generalizes well to the LVIS dataset. ViLD produces accurate masks for cloth and cloth-like objects, and has been implemented with success in other interactive perception pipelines (Wu et al., 2023).

ViLD performs poorly with low resolution images of simulated environments like the 64x64 used by FlingBot. At resolutions of 64x64 and 128x128, ViLD frequently detects no objects at all. This is likely because at low resolutions the cloth appears as a large area of pixels with nearly identical color, something rarely found in real-world images. We provide ViLD with a 256x256 top-down image of a cloth, which retains defining details like wrinkles, resulting in significantly better results.

We provide ViLD with several categories:

```
detection_categories = ["cloth", "clothing",
"fabric', "square", "trapezoid", "shirt",
"towel", "green cloth", "blue cloth", "purple
cloth", "red cloth", "yellow cloth", "white
cloth", "orange cloth"]
```

We provide the categories “cloth,” “clothing,” “fabric,” and “shirt,” to generically detect the cloth in each image. We further provide several categories which include common colors, which often result in higher confidence values than the generic ”cloth” prompt and can detect cloths that would otherwise be classified as background. In cases where the cloth is folded such that it closely resembles a polygon, ViLD often fails to classify it as any of the previous categories. We provide the categories ”trapezoid” and ”square” to detect the cloth in cases where it closely resembles those shapes.

ViLD produces image masks for any instances of these categories found in the image. We select the mask with the highest confidence value for any of the given categories. We eliminate all boxes with an area below 100 pixels to filter out inaccurate masks, and apply non-maximum suppression to further eliminate duplicate masks.

3.2 Angle Value Maps

One way of representing points around the edge of a cloth is by their angle from the centroid. We implement this technique by predicted grasp points with a 360-value dense angle map. In contrast to FlingBot's spatial value map, our angle map reduces the task of grasp point selection from selecting from 4096 points to just 360 which fills the space of possible ideal grasps just as well. This method also prevents FlingWAM from predicting grasp points outside of the cloth when presented with novel tasks.

The two grasp angles are the highest-scored angle, and the next highest scored angle that is at least 25° from the first angle. The grasp points are derived by finding the point on the cloth furthest from the centroid at a given angle. The grasp points are adjusted slightly towards the centroid to avoid real-world grasping failures. The grasp points are switched if necessary to avoid collision of the robot arms in real-world settings

3.3 Self-Supervised Training

Our model is adapted from FlingBot, and can be split into a fully convolutional segment and a DNN segment. The first segment contains a convolutional layer, followed by 6 residual blocks, and finally three further convolutional layers to reduce the number of channels to 1. The output of this model is then flattened and passed into a DNN which reduces the vector representing the cloth into an angle map, consisting of 360 values.

We use FlingBot's self-supervised training method. For each regression step, both chosen angles in an angle map are regressed to the delta-coverage resulting from the fling action. We generate a training dataset of 2000 initial cloth orientations. Our model is trained using the Adam optimizer with a learning rate of 1e-4, reduced to 3e-5 after 60k steps and a weight decay of 1e-6 with a rolling batch window of 64 interactions, regressing after every other simulation step. It is trained on a PC with a single Quadro P5000 GPU until improvement in success percentage stops, after 104,000 simulation steps.

Due to the initial randomized state of the model, early in training, the model regresses each value in the angle map to the average delta-coverage achieved by all flings consisting of that angle. In later stages of training, the model only samples the highest-scored values. This results in each value regressing to the delta-coverage achieved from the best fling consisting of that angle. Because the sampling pattern of the model changes throughout the later stages of training, loss doesn't decrease significantly during that period while the results of the model continue to improve steadily. In contrast to a spatial value map, each value in our angle map is sampled with near-equal frequency, improving training efficiency and allowing for smaller batch sizes.

3.4 Simulation System

Algorithm 1 System pipeline

```
while Cloth Not Unfolded do
    $I_{\text{top}}$ = GetOverheadImage()
    $m$ = $\textit{ViLD}$.GetMask($I_{\text{top}}$)
    $c$ = GetCentroid($m$)
    $am$ = $\textit{FlingWAM}$.GetAngleMap($I_{\text{top}}$, $m$, $c$)
    $g$ = $\textit{FlingWAM}$.GetGrasps($am$, $m$, $c$)
    $\textit{FlingBot}$.PickStretchFling($gl$)
end while
```

We use the same simulation engine as FlingBot; our simulation pipeline code is also adapted from FlingBot. We first take a top-down RGB image of the workstation. The image is cropped and resized to center the cloth, with dimensions of 256x256. This image is first passed into ViLD, which outputs a binary mask. The centroid of the binary mask is calculated and represented as a binary two-dimensional matrix with a value of 1 at the centroid. The RGB image, binary mask, and centroid matrix are all inputs for our model. Because FlingWAM's grasp point selection model only takes 3 images as input compared to FlingBot's 96, the image quality can be increased significantly while still retaining comparable memory usage.

Just as ViLD experiences a significant decrease in accuracy when provided with low-resolution images, a convolutional grasp point selection model requires a similar resolution level to adequately interpret visual properties of cloths. FlingWAM's model takes 256x256 images as input, significantly increasing the ceiling for its success rate in comparison to FlingBot.

As described in Section 3.2, grasp points are derived from the angle map, cloth mask, and centroid. The cloth is then flung and the delta-coverage calculated. The robot stops its interaction with the cloth if any of four cases is met: 1. The cloth reaches maximum coverage, 2. ViLD fails to produce a mask, 3. The model selects grasp points which are not on the cloth due to an inaccurate mask, 4. The model reaches a hard limit of 9 interactions.

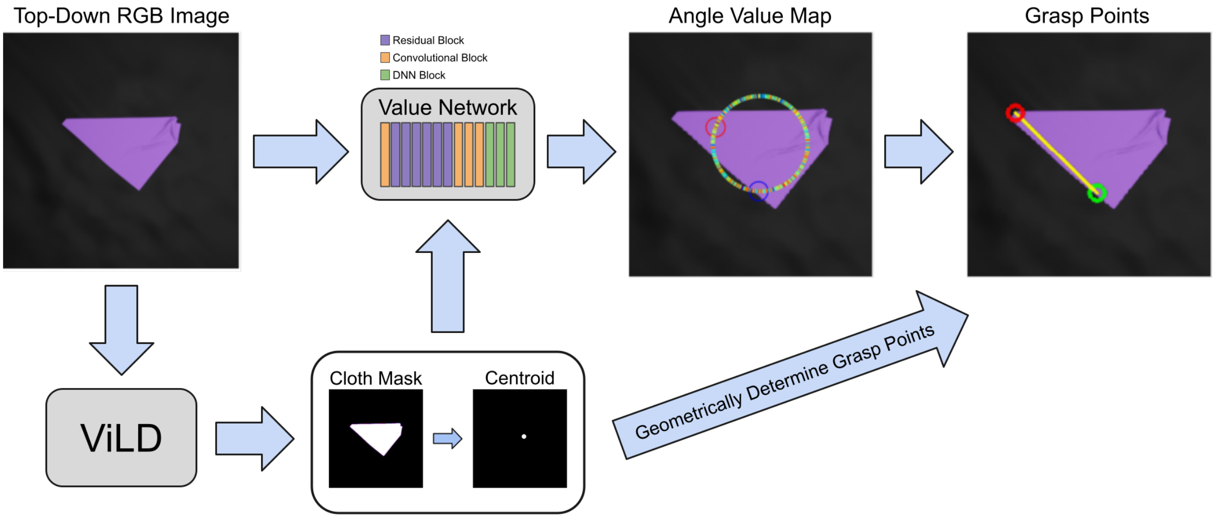

Fig. 1 **System overview.** Beginning with a top-down image of the workspace, we use ViLD to determine a cloth mask and centroid. We provide a value network consisting of a CNN with residual blocks followed by an MLP with the top-down image, mask, and centroid. The value network returns an angle value map densely populated with 360 values. We select the two highest valued angles which are at least 25° apart. We geometrically determine grasp points by projecting lines at the chosen angles from the centroid until they reach the edge of the cloth.

4 Results

We evaluate the efficacy of our model in simulation against the FlingBot baseline. We further analyze the accuracy of the ViLD mask-generation component of our model.

4.1 Baseline comparison

We compare FlingWAM with FlingBot in simulation using a dataset of 800 randomly generated crumpled cloth configurations. The results are in Tab. 1.

- **Coverage**: the area covered by a cloth as a percentage of maximum possible area
- **Final Coverage**: the coverage achieved at the end of a series of interactions with one cloth
- **Best Coverage**: the best coverage achieved at any point during a series of interactions with one cloth
- **Success Percentage**: as defined in section 1, is the percent of final coverages larger than or equal to 1.0.

FlingWAM significantly outperforms FlingBot in success percentage. FlingWAM's average final coverage is significantly lower than its average best coverage, suggesting that FlingWAM frequently makes mistakes that reduce cloth coverage. In contrast, FlingBot's average final coverage is nearly as high as its average best coverage. This is because of FlingBot's previously mentioned ability to end cloth interactions when it perceives that no more improvements can be made.

Table 1 Comparisons to baselines

Method	Success %	Avg. Final Coverage	Avg. Best Coverage
FlingBot	13.3%	**.912**	.935
FlingWAM	**23.5%**	.902	**.951**

4.2 Ablation studies

FlingWAM without feature engineering. In FlingWAM, we manually provide our model with binary mask and centroid features. Because our grasp points are derived from the cloth's centroid, the model must have some understanding of the location of the centroid. However, it has been proposed that deep learning models like CNNs are capable of extracting features automatically (Shaheen et al., 2016). To test this theory, we build a version of FlingWAM which provides the model with only an RGB top-down image of the environment (FlingWAM-img). The cloth mask

Table 2 Comparisons to baselines

Method	Success %	Avg. Final Coverage	Avg. Best Coverage
FlingBot	13.3%	**.912**	.935
FlingWAM	**23.5%**	.902	**.951**
FlingWAM-img	16.8%	.837	.932

and centroid are still used when calculating grasp points from angles. We train FlingWAM-img until improvement in success percentage stops, after 45,000 simulation steps. The results are in Tab. 2. FlingWAM-img performs significantly worse than FlingWAM, suggesting that in this instance, feature engineering is useful at providing FlingWAM with a geometric understanding of the angle maps it is producing.

4.3 Failure Cases

ViLD image segmentation. ViLD produces no mask in roughly 9% of cloth tasks, immediately ending the task. This limits FlingWAM from achieving higher cloth coverage in many tasks. It also sets an upper bound on the possible success of any cloth unfolding pipeline using ViLD or equivalent OVIS models. However, OVIS models may perform better in real-world experiments with stronger similarity to their training data.

ViLD also produces inaccurate cloth masks a non insignificant percent of the time. This can sometimes result in grasp points not on the edge of the cloth or outside of the cloth.

Simulation. SoftGym, FlingBot's simulator, doesn't provide cloth tension data. To work around this, the cloth stretching aspect of FlingBot's fling primitive stretches until the center of the stretched cloth stops moving. This method often results in simulated stretching well beyond the physical limits of most real-world cloths. It also results in significant recoil when the cloth is released at the end of the fling primitive. This decreases the efficacy and real-world accuracy of the fling primitive in simulation.

5 Conclusion and Future Work

In this work, we proposed a method of grasp point selection utilizing image segmentation and angle value maps to select edges for grasping. We demonstrated our model's efficacy in comparison to the FlingBot baseline for selecting double-arm cloth fling grasp points. We motivated completion percentage as the primary measure of cloth flinging success and demonstrated that FlingWAM outperforms other models at achieving this goal. FlingWAM resolves several of FlingBot's failure modes, including the discretization of grasp angles, resolution of workspace images, and stipulations for terminating an interaction. FlingWAM simultaneously maintains comparable memory usage and inference time.

There is still room for improvement in the realms of fling primitive design, image segmentation, and grasp point selection. While FlingWAM represents a significant improvement on previous work, it is still achieves successful flings too infrequently to be useful in a household setting. Future work should attempt to optimize fling actions given information about cloth structure. Image segmentation models trained specifically on cloth datasets would significantly improve the efficacy of our pipeline. Future work should also attempt to implement FlingWAM in a real-world setting.

Acknowledgments

The authors would like to thank Brandon Vu, Jingyun Yang, Kevin Lin, and Huy Ha for fruitful technical discussions. This research was conducted thanks to a sponsorship from Jeannette Bohg with the Interactive Perception and Robot Learning Lab at Stanford University.

References

Ha, H., Song, S.: Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In: Conference on Robotic Learning (CoRL) (2021)

Willimon, B., Birchfield, I. Stan Walker: Model for unfolding laundry using interactive perception. In: International Conference on Intelligent Robots and Systems (IROS) (2011)

Hamajima, K., Kakikura, M.: Planning strategy for task of unfolding clothes. In: Robotics and Autonomous Systems (2000)

Yamazaki, K.: Gripping positions selection for unfolding a rectangular cloth product. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE) (2018)

Wu, J., Sun, X., Zeng, A., Song, S., Lee, J., Rusinkiewicz, S., Funkhouser, T.: Spatial action maps for mobile manipulation. In: Robotics: Science and Systems XVI (2020)

Wang, P., Wang, S., Lin, J., Bai, S., Zhou, X., Zhou, J., Wang, X., Zhou, C.: ONE-PEACE: Exploring One General Representation Model Toward Unlimited Modalities (2023)

Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., Wang, X., Qiao, Y.: InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions (2023)

Gu, X., Lin, T.-Y., Kuo, W., Cui, Y.: Open-vocabulary Object Detection via Vision and Language Knowledge Distillation (2022)

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., Houlsby, N.: Simple open-vocabulary object detection with vision transformers. ECCV (2022)

Kamath, A., Singh, M., LeCun, Y., Misra, I., Synnaeve, G., Carion, N.: Mdetr–modulated detection for end-to-end multi-modal understanding. arXiv preprint arXiv:2104.12763 (2021)

Wu, J., Antonova, R., Kan, A., Lepert, M., Zeng, A., Song, S., Bohg, J., Rusinkiewicz, S., Funkhouser, T.: Tidybot: Personalized robot assistance with large language models. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2023)

Shaheen, F., Verma, B., Asafuddoula, M.: Impact of automatic feature extraction in deep learning architecture. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8 (2016)