Wrinkle-attentive Fabric Manipulation Metric

Runsheng Wang[1], Dongxiao Yang[1], Amr El-Azizi[1], Khalifa AlQaydi[1]
[1]Columbia University, NY, USA

***Abstract*—Garment folding is a popular topic in deformable object manipulation. While most studies use the Intersection over Union (IoU) metric to evaluate the quality of folds, there are many cases where this metric can produce a misleading quantification, especially when wrinkles are present. In this work, we seek to overcome the limitations of IoU by designing an enhanced metric that combines wrinkle-detection scores with Generalized IoU (GIoU). To demonstrate the effectiveness of this metric, we build a simulation environment and train a model to fold a square fabric. We compare the results under our new metric with traditional IoU approaches and demonstrate that our metric is superior in evaluating folding quality. We also show that our metric has more than enough sensitivity when applied to real-world data captured by a off-the-shelf RGB-D camera. Our metric provides a useful way to evaluate deformable manipulation, and it has the potential to be applied to downstream cloth-folding tasks, especially in self-supervised learning.**

I. Introduction

THE ability to handle objects quickly and efficiently is crucial for modern robotics, especially in tasks such as pick and place, grasping, folding, and packing. Many of these tasks are still primarily performed by humans, particularly when manipulating deformable objects like clothes, due to the complex configuration space that requires significant perceptual and non-linear dynamics abilities [1] [2]. For instance, cloth folding is a sequential action which requires microscopic interactions that are difficult to model and plan [3]. Furthermore, standard manipulation strategies are often rendered useless when applied to deformable objects with near-infinite degrees of freedom. Many complexities arise when the object being manipulated slide, shift, crumple, or form unexpected wrinkles [4]. This flexibility is further amplified when the cloth is made of multiple layers of fabric stacked on top of each other.

Although there has been significant progress in building autonomous robotic systems in recent years, manipulation of deformable objects remains challenging as there is limited real world data to leverage [5]. Additionally, there is a lack of reliable quantitative metrics that can be used for deformable object manipulation. This often limits works with deformable cloth to the realm of supervised training and data collection.

In this paper, we intend to carry out an in-depth analysis of folding performance metrics, as well as defining a new metric that combines GIoU and Wrinkle detection as a single objective loss function (Fig. 1). We hope this combined loss can then be used for fully self-supervised training.

The contributions of this paper include:

1) A novel metric design combining wrinkle detection and GIoU into a single loss function.
2) A rotational and shift invariant GIoU algorithm that can be extended to other image processing needs.
3) A simple self-supervised training fold action to demonstrate the metric.

The rest of this paper is organized as below. We first review related literature, before describing the metric definition and techniques to pre-process the RGB-D images of clothes. We then present our results from simulated environments and real-world data. We conclude by summarizing our findings, limitations, and potential future directions.

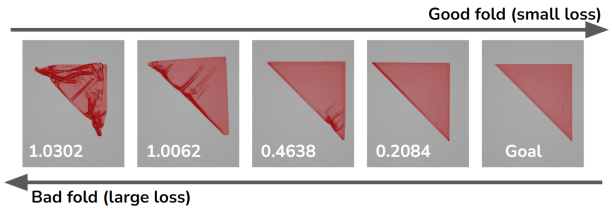

Fig. 1. Loss values of five folding results calculated from using our combined metric. From left to right the folding quality gets better while the loss value gets smaller. The rightmost image is the goal state (0 loss).

II. Related Work

A. IoU Metric

IoU is one of the most popular quantitative metrics used in deformable manipulation tasks. It is used in [5] to quantify approximate progress towards folding goal when the initial conditions were similar. Traditional IoU approaches are often not invariant to rotation and shifts in the goal state. We address this problem by introducing a rotational and shift invariant version of Generalized IoU.

B. Fabric Smoothing

Cloth smoothing is commonly performed through a series of grasping and pulling actions to transform it from highly wrinkled and crumpled configuration to smooth configuration [6]. Most of prior work focuses on learning smoothing policies on a garment by using expert demonstrations [2] or extracting specific features that represent wrinkles and corners, before smoothing the fabric [7]. These approaches treat smoothing and IoU separately, and often require several additional interactions with the cloth to eliminate wrinkles. In this paper, we aim to simultaneously optimize smoothness and GIoU instead of treating them separately, which enables us to properly manipulate the objects from the beginning.

C. Wrinkle Representation

Some wrinkle detection methods include applying Gabor filtering to an intensity image as proposed by [8]. However, such approach limits robot ability to detect wrinkles due to the extreme sensitivity of intensity-based image features [7]. In this paper, we instead employ a height based representation, which is computationally more efficient and makes the data collection process easier.

III. Method

A. Metric Definition

We propose the following metric for evaluating cloth folding:

$$L = \{1-(\frac{A\cap B}{A\cup B} - \frac{|\frac{C}{A\cup B}|}{|C|})\} + \{1-\max(\frac{\frac{\sum_{a\in A} z_a}{P_A}}{\frac{\sum_{b\in B} z_b}{P_B}}, 1)\} \quad (1)$$

A and B are two arbitrary convex shapes (cloths), where A is the goal state and B is an arbitrary state we seek to evaluate. C is the minimum convex hull that encloses A and B. Developed by Rezatofighi et al., the component

$$\frac{A\cap B}{A\cup B} - \frac{|\frac{C}{A\cup B}|}{|C|}$$

is a Generalized Intersection over Union (GIoU) metric [9].

There are many ways to describe wrinkles. For our metric, we utilized height measurements as an representation for smoothness conditions. To encode this information, we first define z_i as the height of the cloth at a particular pixel location i on the 2-dimensional convex shape I. P_A and P_B are the total number of pixels of convex shapes A and B, respectively. We then compute the sum of all the height values separately for goal state A and arbitrary state B, before standardizing the sums by dividing them with P_A and P_B. We divide the standardized sum of height values of the goal state A by that of an arbitrary state B. If the smoothness condition of the arbitrary state is close to that of the goal state, we expect this ratio to be close to 1.

The standardization of the sums through the division of P_A and P_B are necessary and useful for two reasons. First, suppose we are interested in computing this metric for a piece of cloth with two different convex shapes, where A is a wrinkle-free goal state and B is an arbitrary wrinkled state. It is worth noting that the z_i values for any pixel location on any cloth, even for a wrinkle-free goal state, is always non-zero as all cloths have a non-zero thickness. Therefore, it is possible that $\sum_{a\in A} z_a \approx \sum_{b\in B} z_b$, meaning that the wrinkled state has a sum of height values that is close to the goal state's sum of height values (e.g. the wrinkled state has a very small P_B with high z_b values at all pixels). If we computed the ratio without standardizing the sums, then the ratio would be close to 1, which would be misleading. Standardizing by P_A and P_B resolves this problem because pixel areas is being taken into account. Another benefit of this standardization is that it enables this metric to be interpretable for humans. Each standardized sum can be thought of as the "average" level of smoothness across the entire cloth.

We perform a $max()$ operation between the ratio and 1 in order to make the metric more robust to rare scenarios. We assume that the goal state is wrinkle-free. Goal states with a specific wrinkle pattern are beyond the scope of this study. Under this assumption, the standardized sum of heights of the goal state would always be less than or equal to the standardized sum of heights for an arbitrary state. However, it is impossible to guarantee that the goal state is the most wrinkle-free state for a specific folding pattern, as a given state could 'outperform' the goal state by very small margins due to randomness, camera calibration, experimental setup, and other latent variables. Therefore, it is possible for the denominator of the ratio term to be less than the numerator, resulting in a value greater than 1. We would like this ratio to be between 0 and 1, with larger values indicating better folds, so that its scale is comparable to the GIoU term in the metric. Thus, if an arbitrary state B results in this ratio being greater than 1, we would consider B as a "perfect" fold, indicated by a value of 1. Note that as a result of this max function, the wrinkle component is no-longer differentiable, as at the max, the gradient becomes 0. In the future, we'd like to evaluate the metric without the max function, to see if it can be made differentiable without loss of boundedness. Since GIoU is differentiable everywhere, and the sum of two differentiable functions is differentiable, this would allow the Loss function to be used in backpropagation

We formulated this metric as a loss function. Therefore, the value of L should be small when the goal state and the arbitrary state are close to one another. Since both GIoU and the wrinkle component of our metric approaches 1 for similar states, we subtracted each component from 1 to define a meaningful loss function.

Our metric extends prior works on wrinkle detection and cloth folding in several aspects. First, it combines GIoU and smoothness into a single unit-less metric where the scales of GIoU and smoothness are comparable. Secondly, it is bounded between -1 and 2, as GIoU is bounded between -1 and 1 and our metric is bounded between 0 and 1. We demonstrate our method for extending GIoU to be rotation and shift invariant in the following section, which will bound the metric between 0 and 2. Lastly, it offers an efficient way of representing wrinkle information in deformable objects, and data collection is quite easy.

B. Image Pre-processing

Current formulations of IoU and GIoU are not invariant to rotation and shifts. Extending them to be rotational and shift invariant is useful because objects in real-world tasks are often not confined within a particular region. We hope to be able to apply our metric to compare goal states and given states regardless of camera placement and folding orientation. Therefore, we preprocessed input images through a two-step process to account for rotational and shifts when computing GIoU.

Rotational Invariant: To make the metric rotational invariant, we create a set of rotated goal images to compare against our arbitrary stae. We first segment the cloth based

on RGB colors (though other segmentation methods can be used if background color is of concern). We then identify the contours and the minimum enclosing circle (MEC) of the cloth. Using the center of the MEC as the rotational center, we rotate the MEC 359 times, starting from a rotational angle of 1 and increasing the rotational angle by 1 for each subsequent iteration. We save the resulting images into a local folder along with the original goal state image, obtaining a total of 360 possible shifted goal states for a given goal state. This can be expanded to arbitrary granualirty as necessary by decreasig the size of the rotational angle step. Fig. 2 illustrates this process

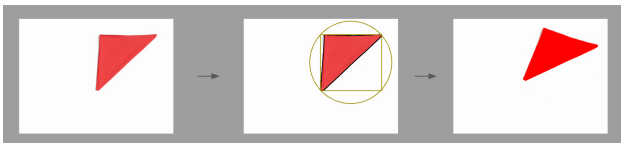

Fig. 2. The minimum enclosing circle is constructed around the cloth, before rotating the MEC 359 times.

Shift Invariant: To make our calculation, shift invariant, we apply a Hill-Climbing algorithm on the arbitrary input state B. We first segment the input cloth and identify its bounding box. We then center the bounding box of the arbitrary state to the center of the MEC of the goal state. From there, we initialize a hill-climber algorithm with a population size equal to the amount of rotated images (in our case, 360), and compute the GIoU loss between this centered test image and each rotated goal image. For each input shifted and goal rotation pair, we then shift the input image by 1 pixel in the four directions (up, down, left, right), rejecting any shifts that would move the cloth out of the image bounds. We then compare the GIoU of the 4 shifted images with the appropriate rotated goal to their initial state. If they show no improvement and the original was not the global minimum, we eliminate that rotation from our potential solutions. Otherwise, we select the shift with the best GIoU for each rotation and iterate. We repeat this process until we're left with a single global minimum that cannot be improved further by shifting.

Note that our current algorithm only makes an image Shift Invariant in the x-y plane, and does nothing to account for the camera shifting depth between the goal state and arbitrary state photos. In the future, we hope to account for this by tallying the z-heights relative to the depth of the surface the cloth is placed on, rather than utilizing the absolute value of z-heights. This can simply be done by identifying the surface behind the contour of the cloth, which we already possess, and subtracting the z-height of the surface either from the heights of the cloth in the raw data or within our metric. A sample equation for this correction is provided below:

$$1 - \max\left(\frac{\frac{\sum_{a\in A} z_a}{P_A} - z_{surface,a}}{\frac{\sum_{b\in B} z_b}{P_B} - z_{surface,b}}, 1\right) \tag{2}$$

IV. Experiment and Result

A. Simulation Environment

We set up our simulation environment using Blender [10]. Blender has a built-in engine that simulates physics, including collision and friction, making it a good simulator for cloth physics. Cloths are simulated as a square plane mesh subdivided into $N \times N$ grids. A larger N gives more delicate deformation in the simulation but significantly slows down the computational speed. Meanwhile, an overly large N is also unreasonable for most real-world cloths. Therefore, we settled on an N of 40 that guaranteed both efficiency and accuracy.

As shown in Fig. 3, the cloth was fully flattened in the initial state. We then represent our simplified robot gripper by pinning one of the cloth corners to a hook that is movable in 3D space. The motions were simulated and animated frame by frame. Given initial location (x_0, y_0, z_0), the "gripper" first moves to $(x_0 + \frac{dx}{2}, y_0 + \frac{dy}{2}, z_0 + dz)$ from frame $F_i = 5$ to frame F_m using a linear path. Then it continues to move to the target position $(x_0 + dx, y_0 + dy, z_0)$ until frame $F_e = 2(F_m - 5) + 5$ through a linear path. Finally the cloth corner is unpinned from the "gripper", and 30 more frames are animated to allow complete settlement of the cloth.

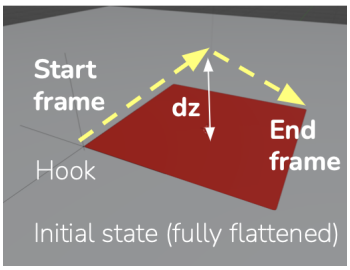

Fig. 3. The cloth folding environment setup in Blender. The robot gripper is represented by a hook entity that follows two linear track during folding, parameterized by the lifting height dz and the end frame when the gripper finishes its motion F_e.

In our study, we only investigate one goal state: the diagonal single fold. Thus we fix the picking location (x_0, y_0, z_0) and placing location $(x_0 + dx, y_0 + dy, z_0)$. The variables to be optimized are the lifting height dz and the speed of the movement parameterized by F_e.

It makes sense to look into these two parameters as they both influence the quality of the fold and can be easily translated using kinematics equations into the motion of an arbitrary robot arm. It also allows us to simplify the problem space to focus on the core-criteria of evaluating the quality of the metric. A few recent studies also investigated the folding trajectory and speed. De Gusseme and Wyffels optimized the lifting height and tilt [11], and Avigal et al. studied to improve the speed of motion while ensuring folding quality [12]. If the lifting height is too low or the speed is too slow, the cloth will wrinkle due to self-friction. If it is too high, the gripper will lift the entire cloth up, resulting in another bad folding. Finally, if the speed is too high, it can not only drag the cloth out of workspace, but also flip the entire cloth upside down.

B. Training with Hill-Climbing Algorithm

To optimize on our two variables, we set up self-supervised training with a Random-Mutation Hill-Climber. We start by generating a population of random speed and height pairs within the bounded limits of our robots dynamics. We simulate

folds with these parameters to get our initial loss. Then, for each generation, we duplicate the top 50% of solutions, then randomly modifies the speed or the height by a small delta. Finally, we simulate a fold using the new parameters. If this fold has lower loss, we replace the old parameters with the new one. We repeat this until we hit the limit of generations set at execution start.

We run this for 3 generations with a population size of 5 for the first run and 10 for the second run (a total of 15 and 30 evaluations respectively). We then repeat both of these runs with an ablation on our wrinkle metric, maximizing on GIoU as a loss function alone. The following is a sample of the best results from each run:

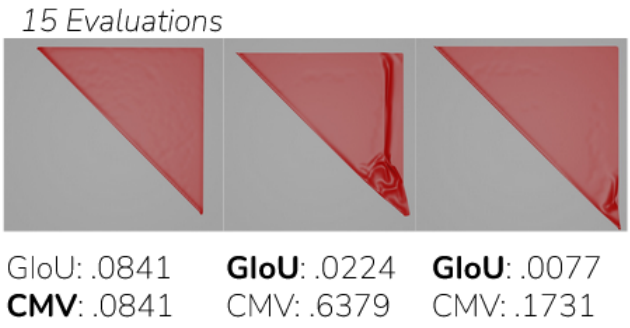

Fig. 4. The top folds from the 15 Evaluations run. Presented are both the GIoU loss and the Combined Metric Value (CMV) Loss. The bolded parameter is what it was trained on, and the other loss is provided for comparison. We include multiple GIoU folds to show the variety of wrinkles in acceptable results.

As can be seen in Fig. 4, training on GIoU provides consistently incredibly small loss with folds that we would consider bad as a result of wrinkles. Meanwhile, for the same duration, our metric creates folds that are incredibly smooth while also having small GIoU loss. In fact, in the first fold, the GIoU and CMV have the same value, which means the wrinkle loss is 0. This means the fold is within an acceptable amount of wrinkles relative to the goal state ground truth, and the ratio is being bounded by the max function at 1.

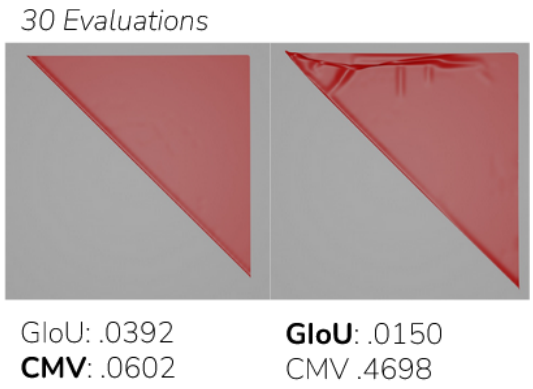

Fig. 5. As with Fig. 4, but for the 30 Evaluation runs. It shows how further evaluations increased the optimization on CMV and GIoU for the CMV Loss run, but minimal returns were provided on the GIoU run.

C. Real World Granularity

One concern with our metric is that real-world depth data would not have the granularity to effectively calculate the wrinkle component. To verify its efficacy, we collected a series of RGBD images of a cloth after a simple triangle fold with various degrees of success, as well as one ground truth fold and one unfolded photo. We then contoured the cloth and calculated the z-heights ratio as per our metric.

(a) 1.0 (b) .9973 (c) .9640 (d) .9219 (e) .8046

Fig. 6. Five real world depth-images and their associated depth ratios as compared to image (a). They're presented in order of increasing loss.

First, we can see that the unfolded cloth was still quite wrinkled, so despite being the best performing it didn't quite score 1. This also reinforces that GIoU is still important to have in the metric, otherwise an unfolded state would be considered the result that minimizes loss. Second, we can see that as the amount of wrinkles or height of the fabric increases, the value of the ratio accurately decreases. Therefore, a preliminary examination proves that real world depth data does have the necessary granularity to further pursue this metric. It also worth noting that the bounded deviation of these values provides further optimism that removing the max to make the function differentiable will likely be successful.

V. Conclusion

We have proposed a new metric for evaluating the quality of folds that combined GIoU and wrinkle detection. We showed that it is a bounded function that is generalizable to arbitrary fabrics and goal-states, as long as it has access to a ground-truth goal state, a limitation it inherits from GIoU. We demonstrated that in simulation it shows increased quality and speed of fold optimization when compared to GIoU for a simple task, and that real world data has the granularity necessary for exploring further sim-to-real transfer.

VI. Discussion

There's still a lot of work to be done with regards to first verifying the efficacy of the metric, then beginning to apply it to more complex problems.

First, we would like to carry out human verification, where we compare if humans and our metric will agree on the better fold. We can do this through random sampling of pairs of folds and seeing which ones humans would consider to be a better fold. Then we would compare the results with GIoU, Wrinkle only, and CMV, to see which had the highest agreement with humans. Additionally, we'd like to evaluate our metric with the proposed modifications of removing max and making it Z-shift Invariant.

Next, we would also like to ablate GIoU from our simulation and only run utilizing wrinkle to see the results. Additionally,

we would like to run our simulation over longer evaluations to see if GIoU would produce comparable folds over time or if our metric would continue to outperform, as well as gather data such as a convergence plot to see which has a better speed of training. We could further test utilizing gradient descent or simplex pattern search algorithms that we deemed were unnecessary with the quality of results we already received, but could be valuable over longer runs.

Additionally, we would want to simulate with an increased diversity of problems, such as adding more optimization parameters, varying friction and cloth properties, introducing more complex folds, etc.

If our metric succeeds under all the above conditions, we believe it would be ready to apply to more traditional and complex deformable material problems. We would hope to see real-world robot training, the metric being combined with other tasks such as self-supervised pick and place training, and finally further exploring more complex tasks and goal states.

REFERENCES

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine* , vol. 29, no. 3, pp. 67–77, 2022.

[2] A. Ganapathi, P. Sundaresan, B. Thananjeyan, A. Balakrishna, D. Seita, J. Grannen, M. Hwang, R. Hoque, J. E. Gonzalez, N. Jamali et al., "Learning dense visual correspondences in simulation to smooth and fold real fabrics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)* IEEE, 2021, pp. 11 515–11 522.

[3] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlavá ˇc, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[4] P. Jiménez, "Visual grasp point localization, classification and state recognition in robotic manipulation of cloth: An overview," *Robotics and Autonomous Systems* , vol. 92, pp. 107–125, Jun. 2017.

[5] R. Lee, D. Ward, A. Cosgun, V. Dasagi, P. Corke, and J. Leitner, 'Learning Arbitrary-Goal Fabric Folding with One Hour of Real Robot Experience', *CoRR*, vol. abs/2010.03209, 2020.

[6] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali et al., "Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9651–9658

[7] K. Li, G. Camatasa, P. Cockshott, and S. Rangers, "A heuristic-based approach for flattening wrinkled clothes," Aug-2013. [Online]. Available: *http://dx.doi.org/10.1007/978-3-662-43645-5_16.*

[8] Yamakazi, K., & Inaba, M.: A cloth detection method based on image wrinkle feature for daily assistive robots. In: *IAPR Conference on Machine Vision Applications*, pp. 366-369. (2009)

[9] Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression, Dec 2022, *https://giou.stanford.edu/GIoU.pdf*

[10] *Blender 3.4 Python API Documentation*, Dec 2022, https://docs.blender.org/api/current/index.html.

[11] V, De Gusseme and F, wyffels, "Effective cloth folding trajectories in simulation with only two parameters," *Frontiers in Neurorobotics*, Vol.16, 2022, DOI=10.3389/fnbot.2022.989702, https://www.frontiersin.org/articles/10.3389/fnbot.2022.989702.

[12] Y, Avigal, Yahav, L, Berscheid, T, Asfour, T, Kröger, and K, Goldberg, "SpeedFolding: Learning Efficient Bimanual Folding of Garments," *arXiv*, 2022, DOI=10.48550/ARXIV.2208.10552, https://arxiv.org/abs/2208.10552.