UnfoldIR: Tactile Robotic Unfolding of Cloth

Remko Proesmans[1], Andreas Verleysen[1] and Francis wyffels[1]

***Abstract*—Robotic unfolding of cloth is challenging due to the wide range of textile materials and their ability to deform in unpredictable ways. Previous work has focused almost exclusively on visual feedback to solve this task. We present UnfoldIR ("unfolder"), a dual-arm robotic system relying on infrared (IR) tactile sensing and cloth manipulation heuristics to achieve in-air unfolding of randomly crumpled rectangular textiles by means of edge tracing. The system achieves >85% coverage on multiple textiles of different sizes and textures. After unfolding, at least three corners are visible in 83.3 up to 94.7% of cases. Given these strong "tactile-only" results, we argue that the fusion of both tactile and visual sensing can bring cloth unfolding to a new level of performance.**

***Index Terms*—Force and Tactile Sensing, Sensor-based Control, Dual Arm Manipulation**

I. Introduction

Manipulation of deformable objects breaks fundamental assumptions in classical robotics, such as rigidity, low-dimensional state space, and known dynamics models [1]. This necessitates breakthroughs in robot dexterity, sensing, planning and control, all open-ended problems in the state-of-the-art. Cloth manipulation has emerged as a major test bench for new approaches to these issues [2], being notoriously difficult due to the great variability in textile materials and their immeasurable capability to deform. Because of this complexity, many manipulation pipelines expect the cloth to be unfolded beforehand, so that at least the initial configuration is known. Unfolding itself has been attempted previously using almost exclusively computer vision [3]–[8].

Early work [3] relied on simple perceptual cues such as the height and silhouette of the cloth when held, performing iterative grasping, dragging and lifting actions to unfold it. More recently, FlingBot [4] was able to visually detect two grasp points on the cloth and perform a fling action, thus achieving ~80% garment coverage within three iterations. In SpeedFolding [5], this is expanded upon by adding action primitives, such as dragging, to achieve smoother unfolding. In contrast, Doumanoglou [6] aims for in-air unfolding by directly identifying pre-defined grasp points on hanging garments. In-air unfolding has the benefit of not constraining the surface upon which the crumpled cloth is found in terms of friction, shape or size. Gabas [7] looks for more general features such as cloth edges and corners, hoping to omit cloth identification steps. Li [8] pushes beyond camera-only sensing and uses a single infrared (IR) range sensor to locally optimise a visually selected grasping point.

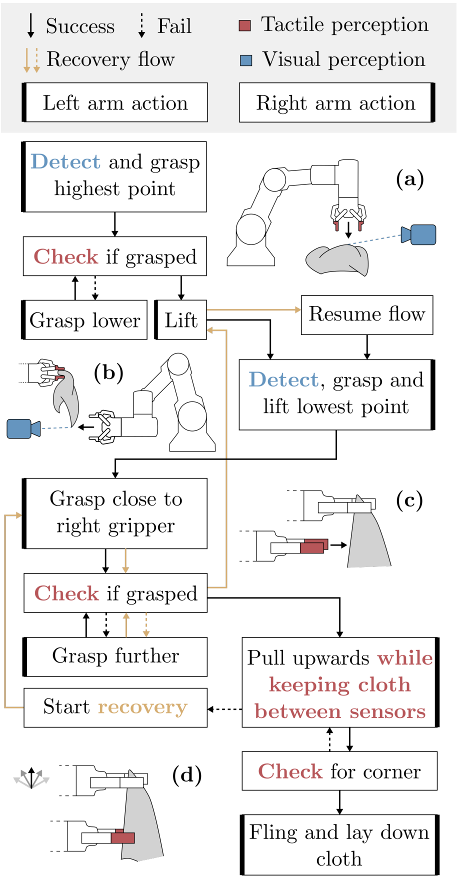


Fig. 1: Unfolding pipeline. Black arrows indicate normal flow. If recovery is initiated, the yellow arrows are followed until normal flow is resumed. **(a)** Grasping highest point of crumpled cloth. **(b)** Grasping lowest point (i.e. first corner) of hanging cloth. **(c)** Setting up edge tracing. **(d)** Tracing edge until adjacent corner is found or cloth slips away.

Generally, tactile sensing finds limited use in robotic cloth unfolding. This despite its potential to solve issues like detecting contact loss [1], which FlingBot suffered from. Robotic tactile sensing has, however, been employed for e.g. texture recognition [9], singulating layers of cloth [10] and cable manipulation [11]. Fusion of both visual and tactile sensor modalities is regarded as a promising avenue for further research [1], [12].

When humans unfold cloth, they rely on their sense of touch. Typically, a person will visually find a first corner and grasp it, and from there, they blindly trace along the edge of the cloth with their second hand, to the next corner [13].

This work was supported by the Research Foundation Flanders (FWO) under grant agreement no. 1S15923N and by the euROBIn Project under EU grant number 101070596.

[1]Remko Proesmans, Andreas Verleysen and Francis wyffels are with the IDLab-AIRO research lab at Ghent University – imec, Technologiepark-Zwijnaarde 126, 9052 Zwijnaarde, Belgium `remko.proesmans@ugent.be`

Inspired by human-like edge tracing, we have developed UnfoldIR ("unfolder", Fig. 1), a dual-arm robotic unfolding procedure shifting focus almost entirely to IR tactile sensing, rather than visual techniques. By introducing more extensive tactile sensing to robotic unfolding, we pave the way towards the integration of both robust tactile and visual modalities for efficient and effective cloth manipulation.

II. Unfolding procedure

Fig. 1 outlines our unfolding procedure. To initialise unfolding, humans have no issue visually finding a first corner on a crumpled cloth. Doing so using computer vision, however, requires advanced techniques and extensive data collection. Instead, we rely on a heuristic described in [7]: when a piece of cloth is grasped randomly and lifted, the lowest point is likely to be a corner. This heuristic works best when minimising the grasped volume of the cloth, such that only one layer or one fold is held. Hence, step (a) in Fig. 1 is to lightly grasp the highest point of the crumpled cloth. The highest point is detected by scanning the depth view of a colour and depth (RGB-D) camera and selecting the highest point within a predefined depth range. The robotic arm grasping this point is augmented with tactile sensors, allowing it to sense the contours of the object it is grasping. In step (a), these tactile sensors are used to detect if the crumpled cloth was successfully grasped at all. The cloth is then lifted, and in step (b), the same RGB-D camera is used to detect the lowest-hanging point, which is assumed to be a corner and grasped by the second arm. The lowest-hanging point is again detected by scanning the depth image, now selecting the lowest point within a predefined depth range. The first gripper releases its grasp, dropping the cloth over one side of the second gripper. This drop has a high probability of exposing an edge close to the second gripper, which is still holding the cloth. In Fig. 1c, the first gripper moves closely below the second, grasping this exposed edge. Subsequently, in step (d), the second arm moves upwards, making sure the edge remains midway between the sensorised fingers of the first gripper.

The drop procedure in step (c) possibly leads to the first arm grasping a fold instead of an edge. However, during tracing, a fold is likely to push itself out from between the fingertips due to the stiffness of the cloth. This can be detected by the sensor fingers. At this point, a recovery procedure is initiated: the first arm regrasps the cloth close to the second gripper, the second gripper releases and the normal unfolding flow restarts at step (b). Edge tracing is successful if a second corner of the cloth passes between the first gripper, as detected by the sensors. The unfolding procedure finalises with a single fling movement to lay down the unfolded cloth.

The edge tracing step involves three proportional controllers, illustrated in Fig. 2. Their gains, K_w, K_α and K_p, are manually tuned. The first controller handles the grasp width w of the first gripper. The edge grasp should be soft enough so that the cloth can slide upwards, while maintaining physical contact between cloth and gripper to prevent the cloth from swinging. However, the grasp width cannot be preprogrammed, as the thickness and texture of the textile are unknown. To account for this, the force torque (F/T) sensor of the first robotic arm is used: if its torque reading τ_i during edge tracing at time instant i notably (>2%) differs from its torque reading τ_0 at rest, there is too much friction between the fingers and the cloth, meaning the first gripper should open by a step w_δ (see Fig. 2). The grasp width updates at a frequency of 20 Hz. At time instant $i+1$, the grasp width w_{i+1} is determined from the previous grasp width w_i by:

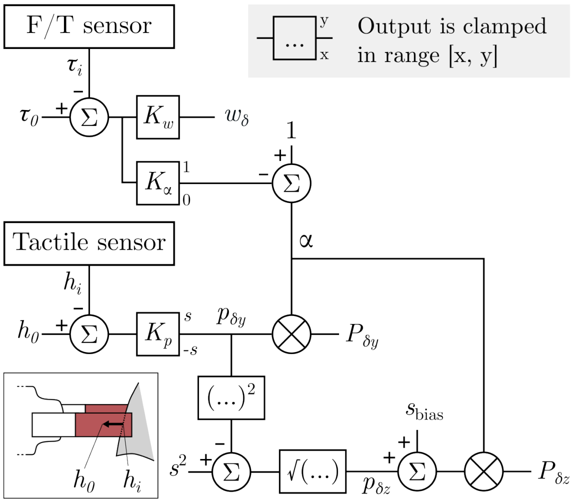


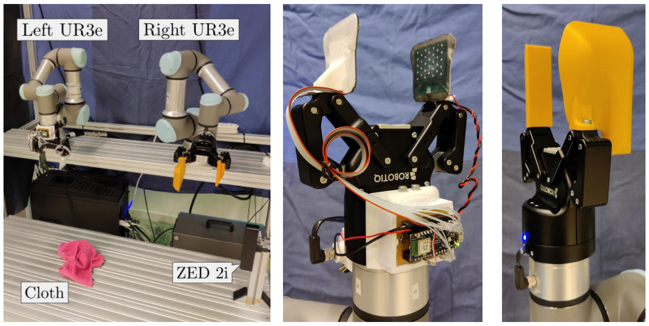
Fig. 2: Edge tracing controller diagram. The bottom left inset clarifies the meaning of h_0 and h_i. Red coloring indicates tactile perception, as in Fig. 1.

$$w_{i+1} = w_i + w_\delta \tag{1}$$

The second controller also employs the F/T sensor: it produces a value α between 0 and 1, which scales the movement step of the second robot arm. If the difference between τ_0 and τ_i is large, the second arm is strongly pulling the cloth, indicating the cloth could be stuck. In this scenario, α decreases towards 0, so that the second arm moves more and more slowly and does not damage the sensor fingers. The third controller keeps the position h_i of the cloth contour, measured along the longitudinal dimension of the sensor fingers, at about midway between the fingers (position h_0). The bottom left inset in Fig. 2 further illustrates h_i and h_0. The controller first applies a position step $p_{\delta y}$ in the horizontal y-direction. If this step is smaller than a predefined maximal step size s, a position step $p_{\delta z}$ in the vertical z-direction is applied such that $s^2 = p_{\delta y}^2 + p_{\delta z}^2$. Furthermore, a constant step s_{bias} is added to $p_{\delta z}$, so that the movement of the second arm always has an upwards component, which makes for smoother edge tracing. The resulting y and z steps are multiplied by α, obtaining $P_{\delta y}$ and $P_{\delta z}$, respectively. The xyz-position $\vec{P}$ of the second robot arm updates at a frequency of 50 Hz. At time instant $i+1$, the position $\vec{P}_{i+1}$ is derived from the previous position $\vec{P}_i$ as follows:

$$\vec{P}_{i+1} = \vec{P}_i + \vec{u}_y P_{\delta y} + \vec{u}_z P_{\delta z} \tag{2}$$

with $\vec{u}_y$ and $\vec{u}_z$ unit vectors in the y and z directions.



(a) System overview. (b) Left gripper. (c) Right gripper.

Fig. 3: Dual-arm robotic setup.

III. Hardware

A. System overview

Our robotic unfolding setup, shown in Fig. 3, consists of two UR3e collaborative robotic arms, each equipped with a Robotiq 2F-85 gripper. Both grippers have been augmented with custom fingertips. On the left arm, the fingertips use infrared circuitry to sense whether anything is placed between them and what its two-dimensional profile looks like. Their design is outlined in section III-B. The fingertips on the right arm are rigid 3D-printed structures. The size and shape of the larger, "club", fingertip is such that a cloth hanging over it will spread out and be more likely to have an edge easily accessible for tracing. The cloth is laid on a working surface 40 cm below the base of either robotic arm. On the right side, a ZED 2i RGB-D camera is mounted 10 cm above the working surface to track the highest or lowest point of the cloth as needed. A second ZED 2i camera is placed in a top-down view. It is only used for randomisation of crumpled cloths (section IV), not during unfolding.

B. Tactile sensing for edge tracing

1) Electrical sensor design: The tactile fingers are two complementary designs, each containing a printed circuit board (PCB): one finger emits IR light, the other receives it. Fig. 4 shows their simplified schematics, along with the components needed for their integration on the left UR3e arm. The first finger contains a grid of 32 IR light-emitting diodes (LEDs), connected in four parallel branches of each eight LEDs. This grid is supplied by a low-dropout voltage regulator, converting the 24 V tool output from the UR3e arm to 10.8 V, which makes for a current consumption of 52 mA after a warmup time of five minutes. The second finger is built around 32 IR photodiodes. Here, four demultiplexers (DEMUX) are included, each addressing eight photodiodes. The DEMUX are all simultaneously controlled by three digital pins of an external Arduino Nano 33 BLE, such that they each route one of four analog pins to one of eight of their respective photodiodes. This finger consumes about 1 mA of current from the 3.3 V output of the Arduino, which in turn receives its power from the 24 V UR3e tool output converted to 5 V by a switching regulator. These electrical specifications are summarised in Table I. The total current consumption reported includes the voltage regulators and the Arduino during operation.

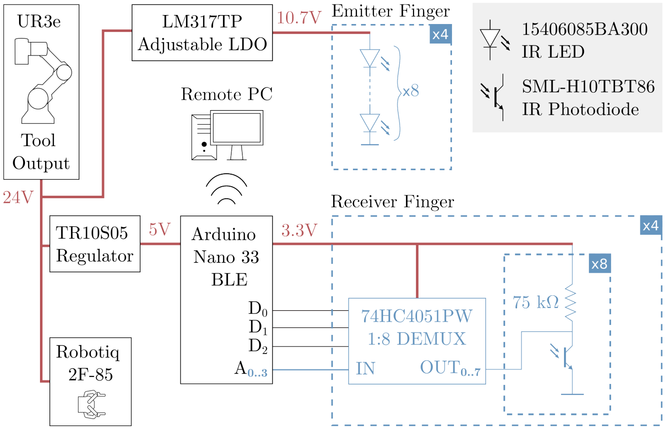


Fig. 4: Electrical diagram of the tactile fingers integrated on a UR3e.

TABLE I: Technical specifications of the tactile fingers.

Characteristic	Value
Total current consumption	60 mA
Receiver finger	1 mA
Emitter finger	52 mA
Warmup time	5 min
Physical size (width × height × depth)	58.0 × 54.0 × 21 mm^3
PCB dimensions	42.8 × 43.1 mm^2
Sensorised area	~38.0 × 30.0 mm^2
Thickness excluding gripper coupling	4.0 mm
Sensor density	3.21 cm^{-2}
Data resolution	8 bit
Readout frequency	55 Hz

2) Structural sensor design & manufacturing: Structurally, both tactile fingers are largely similar. Fig. 5 shows their constituent parts. The IR LEDs and photodiodes are each placed in an identical hexagonal grid on their respective PCBs, such that a single LED is located straight across from a single photodiode when both fingers are brought close together. For each finger, the PCB is slotted into a polyethylene terephthalate glycol (PETG) backing, 3D printed using a Prusa MK3 i3. This combined structure is then placed PCB side down into a mould filled with liquid Silicone Addition Colorless 5 by Silicones and More. After curing at room temperature for 24 hours, the finger is removed from the mould and a 1 mm layer of translucent silicone now protects the PCB components. However, the surface of this silicone layer generates too much friction with common textiles. To solve this, a sheet of 100 µm thick Bemis 3914 thermoplastic polyurethane (TPU) is shaped around the combined PETG, PCB and silicone structure by pulling the TPU taut and heating it locally using a CIF 852 hot air gun set to 90 °C. This manufacturing step is the basis for selecting PETG for the backing instead of polylactic acid (PLA): PETG warps less during heat treatment. Additionally, before shaping, a hole pattern is printed in black on the inside of the TPU layer using an HP LaserJet P2015. Each

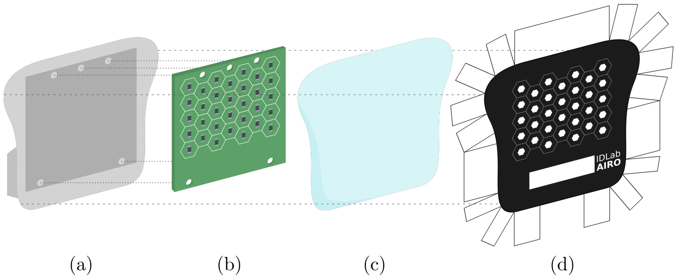
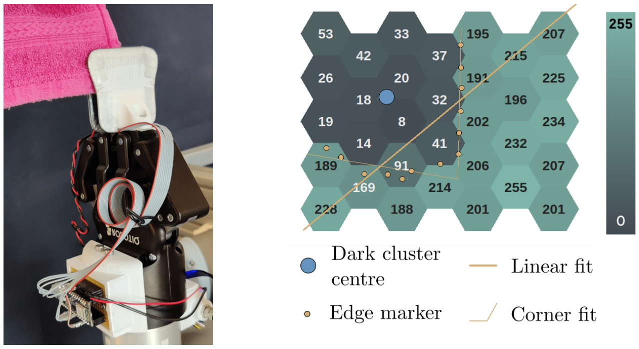


Fig. 5: Sensor structure for both the emitting as well as the receiving finger. **(a)** 3D printed PETG backing. **(b)** PCB, either with 32 IR LEDs or 32 IR photodiodes. **(c)** 1.5 mm thick layer of translucent silicone. **(d)** Sheet of 100 µm Bemis 3914 TPU patterned with black ink.



(a) Corner grasped by tactile fingers.

(b) Sensor readout. The colour scale indicates arbitrary units from minimum (0) to maximum (255) measurable IR light intensity.

Fig. 6: Sensor readout when grasping a textile corner.

hole is positioned directly above an IR emitter or receiver, thus focusing their respective radiation patterns. The entire fingertip is 4 mm thick, making the finger suitable for sliding underneath cloth for grasping, as opposed to e.g. the GelSight sensors used in [11]. Table I provides additional structural specifications.

3) Sensor readout: Getting a single readout from the tactile fingers entails sequentially setting the common selection inputs for the demultiplexers to one of eight codes and reading the voltage on the four analog pins that are as such routed along the IR receiver grid. This data is communicated in an 8-bit format at about 55 Hz to a remote workstation by the Arduino using Bluetooth Low Energy (BLE), as indicated in Fig. 4. When the fingers are close together, the photodiodes receive high IR intensity and pull down the analog pins to ground. Grasped objects between the fingers block a large part of the IR light from the LEDs, which leads to the photodiodes conducting less and results in higher voltages read by the analog pins.

4) Data processing: The grid of 8-bit values received from the tactile fingers must be distilled into a set of parameters in order to use the sensor for closed-loop control. Each value corresponds to a hexagonal cell, of which the position in the grid is defined by its centre coordinates. During unfolding, the grid will either be fully bright, fully dark, or show a combination thereof in two distinct zones. An example of the latter is shown in Fig. 6. The following processing steps are applied to the sensor data:

- The threshold λ deciding which values are "bright", respectively "dark", is calculated by ordering all grid values from lowest to highest and finding the largest step between adjacent values in this ordered list. Any values found before this step are "dark", the others "bright". λ is then set to the average of the bright cluster mean and the dark cluster mean value. Should the bright and dark cluster mean be negligibly close together, it is determined that either the grid is fully bright or fully dark. In this case, further processing steps are skipped.
- The centre point of the dark cluster shows whether or not the cloth is slipping from between the fingers. It is calculated as the weighted mean of the centre coordinates of all cells belonging to the dark cluster, where the weights are the 8-bit complements of the associated values.
- The shape of the border between the dark and the bright cluster indicates whether or not a corner is grasped. To find it, the value of each cell is compared to those of the neighbouring cells. If λ is crossed between cells, the values of the cells are linearly interpolated along the line segment connecting the cell centre points and an edge marker is placed where an interpolated value of λ is found.
- A linear function as well as a piecewise linear, "corner", function are fitted to the edge markers, parameterising the border shape.

IV. Experimental design

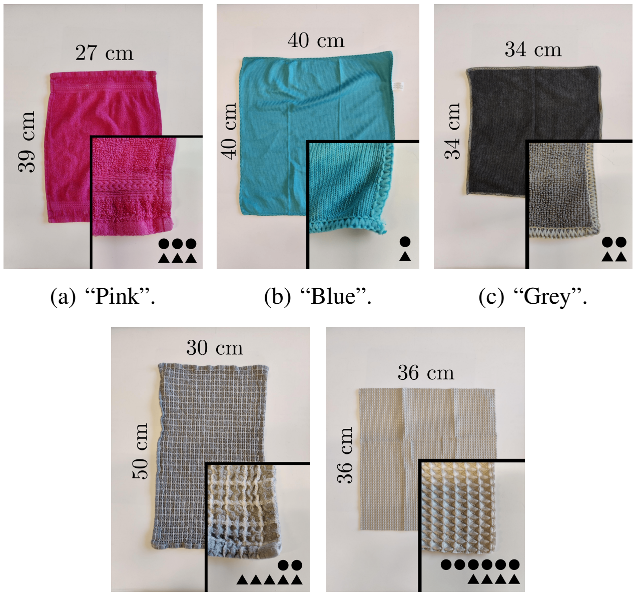
A. Testing cloths

UnfoldIR is tested on five rectangular cloths of different sizes, textures and thicknesses. These are shown in Fig. 7 along with their dimensions and tags by which they will be referred to. Additionally, their relative stiffness and surface roughness is indicated: the more dots, the stiffer the cloth, and the more triangles, the rougher the cloth. These parameters have been qualitatively determined. Roughness is defined as felt by human touch, stiffness by resistance of the cloth to folding. The latter is thus dependent on both cloth material and size. Note the large difference in stiffness between the Beige sample and the others.

Printed text labels have been removed as they can be detected as the lowest-hanging point and confuse the system. They can also be detected as a corner, but most labels are in fact sewn near a cloth corner so this is less of an issue.

B. Random initialisation

It is often neglected in robotics to formalise how random cloth states are achieved. We adhere to the following randomisation methodology. A rectangle the size of the sample is drawn on a top-down video feed of the work area. The angle



(a) “Pink”. (b) “Blue”. (c) “Grey”.

(d) “Chequer”. (e) “Beige”.

Fig. 7: Test samples with dimensions, designated tag, and indication of relative stiffness (dots) and roughness (triangles). The more dots/triangles, the stiffer/rougher the cloth.

of the rectangle is a uniform random value in the [0°, 180°[interval. Additionally, a uniform random point in the inside area of the rectangle is indicated. The test sample is then moved to align with the rectangle on the video feed, after which the point is pinched with two fingers, the cloth is lifted about 10 cm above the work surface, and dropped. We do not claim this method to be general in the sense that it can achieve any possible “crumpled” cloth state, we rather advocate for more standardisation in the field of robotics and maintain that this randomisation methodology is both sufficiently general and reproducible.

C. *Evaluation*

We run 20 trials for each sample. A trial can fail in one of two ways: the cloth is dropped by either one of the robotic arms before the final fling, or the recovery flow is triggered for a third time. Trials can thus be completed without triggering the recovery flow (one-shot completion) or by triggering it once (two-shot completion) or twice (three-shot completion). Experiments that terminate early due to a motion planning error are ignored to not detract from the core concepts of this paper, as also done in [5].

Trials brought to completion are evaluated by three metrics. The first metric measures whether edge tracing has succeeded. Edge tracing is successful when both grippers hold adjacent corners of the cloth between their fingertips before laying it down. Imperfect edge tracing will lead to partial unfolding, which is still considered to be a completed trial. The final coverage of the unfolded cloth is reported as a second performance metric for comparison with other work. Third, we record the number of corners that are visible on the work surface after unfolding. Sufficient visible cloth corners allow subsequent folding procedures relying on keypoint detection [14], [15]. A corner is “visible” if it is not folded onto the cloth itself and locally shows a clear ~90° angle from a top-down perspective. Having all four corners visible is ideal, though having three visible corners still allows for subsequent folding by first grasping two corners and dragging or lifting the cloth.

V. Results

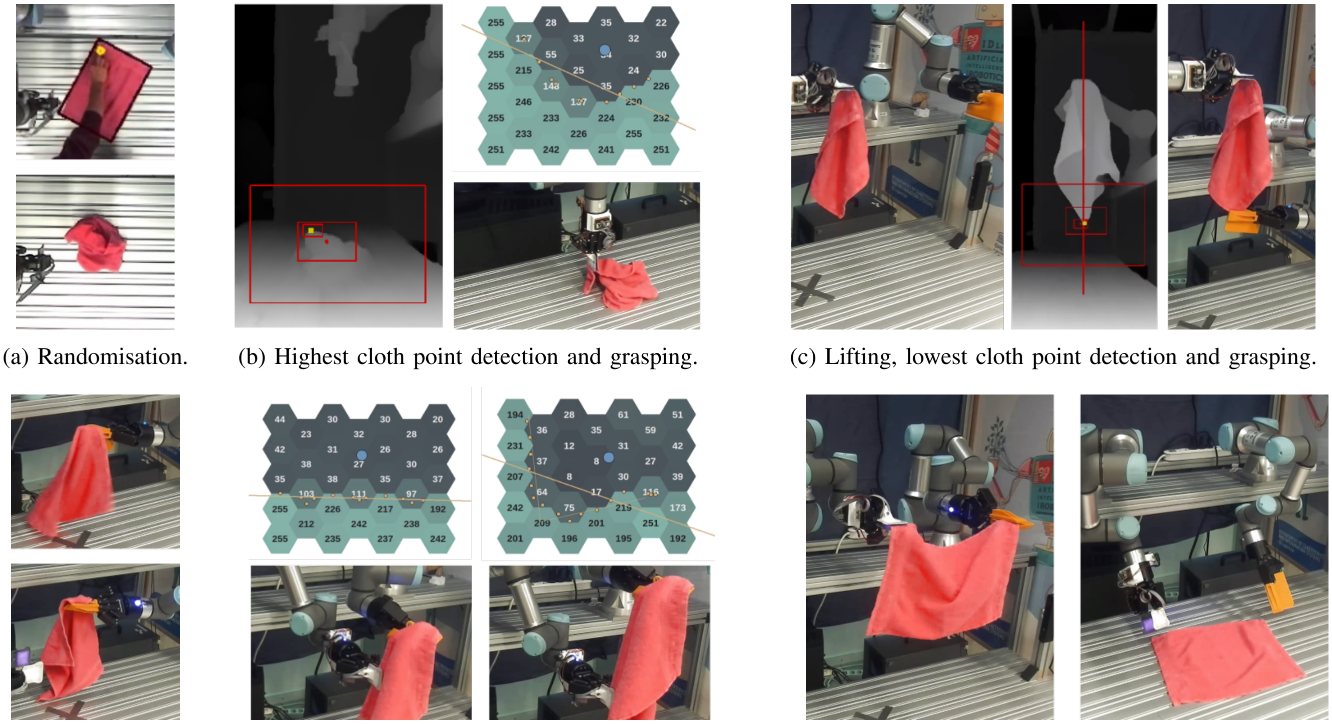
A. *UnfoldIR implementation*

Fig. 8 shows a one-shot trial of the UnfoldIR pipeline on the Pink sample. The randomisation methodology as explained in section IV-B is illustrated by Fig. 8a. The highest and lowest point detection of Fig. 8b and Fig. 8c is achieved by recursively sampling a diminishing region of the current depth image, indicated by the red rectangles, on an increasingly dense grid. After the lowest point of the hanging cloth is grasped by the right arm, the left releases its grasp, dropping the cloth over the narrow finger of the right gripper (see Fig. 3c). The right gripper then rotates 360°, rolling the club finger into the cloth and thereby spreading it out. The arms now move to predefined positions and the left arm grasps forward until it senses the cloth between its fingertips, at which point edge tracing can begin. During edge tracing, the grasp width controller ensures the cloth can easily slide between the sensorised fingertips without uncontrolled swinging, regardless of cloth thickness and texture. When a corner is detected by the tactile fingers, the cloth is spread out in the air and laid down with a single fling movement. We refer the reader to supplementary material for a video showing a set of trials.

B. *Trial run outcomes*

For all five test cloths, Table II contains the number of trials brought to completion, the outcomes of the employed evaluation metrics for the completed trials as well as an indication of what caused trials to fail. The number of completed and failed trials is given as a percentage of the total, i.e. 20. For the tracing success and corner visibility metrics, the percentage of completed trials that satisfy them is reported. The final coverage metric is obtained by averaging the surface area of the unfolded sample over all completed trials and dividing it by the surface area of the sample when fully unfolded. In addition, the standard deviation on these averages is given. The failure cases are grouped into three categories. First, in some cases the right arm fails to grasp the lowest point of the hanging cloth, either due to inadequate visual detection or because the corner of the cloth hung in such a way that closing the gripper pushed it out of the way, rather than grasping it. Second, the cloth can get snagged or dropped in other parts of the pipeline. This comprises a set of failures that each on their own occur rarely: the cloth slips during the fling movement, it gets stuck on a rough edge of the right gripper (Fig. 3c), etc. Third, the recovery flow failed twice, meaning the trial is manually stopped.

A one-shot completion typically takes little over one minute, though the system is not yet optimised for speed. A run through the recovery procedure up until completion, or until a subsequent recovery trigger, takes about 1 minute and 20 seconds due to extra overhead in robot movements.



(a) Randomisation. (b) Highest cloth point detection and grasping. (c) Lifting, lowest cloth point detection and grasping.

(d) Roll. (e) Edge tracing to adjacent corner. (f) Laying down unfolded cloth.

Fig. 8: One-shot trial of the UnfoldIR pipeline on Pink sample.

TABLE II: Experimental results. Values in grey are percentages of the total number of trials per sample (20), values in white are percentages of completed or failed trials.

	Pink	Blue	Grey	Chequer	Beige
One-shot completion	75.0	75.0	65.0	70.0	45.0
Two-shot completion	80.0	80.0	85.0	90.0	55.0
Three-shot completion	95.0	80.0	90.0	90.0	55.0
Edge tracing success	80.0	21.1	60.0	55.0	30.0
Three corners visible	94.7	93.8	94.4	83.3	100
Four corners visible	84.2	31.3	33.3	61.1	81.8
Average final coverage	93.5	85.3	87.5	91.4	99.0
	±12.8	±8.4	±13.6	±17.8	±2.3
Failure cases	5.00	20.0	10.0	10.0	45.0
Lowest point grasp failed	0.0	50.0	50.0	0.0	55.6
Dropped or snagged	100	50.0	0.0	50.0	33.3
Two failed recoveries	0.0	0.0	50.0	50.0	11.1

VI. Discussion

The Pink, Blue, Grey, and Chequer samples show high trial completion rates (80 - 95%, similarly to [7]). Completed trials show high average final coverage (>85%, FlingBot [4] reports coverages >80%) and good results in terms of corner visibility (83.3 - 94.7% show three corners or more). Comparing these four samples in terms of edge tracing success shows that the stiffer the cloth (see Fig. 7), the more likely it is for edge tracing to succeed. The edges of the Pink sample are highly likely to be available for tracing. The Blue sample, on the other hand, has a higher tendency to fold in on itself, so that an adjacent corner is rarely found. In most cases, however, a fold close enough to the edge is grasped such that the tactile fingers still detect a corner when this fold is traced all the way down, leading to a completed trial with three visible corners and good final coverage. The very stiff Beige sample shows excellent scores in terms of coverage and visible corners, because stiff cloths inherently resist deformations. However, only 11 out of 20 trials were brought to completion. In most failure cases, the right gripper failed to grasp the lowest point of the hanging cloth, because the corner was pushed upwards rather than staying between the closing fingers. Furthermore, the Beige sample gets in the way of preprogrammed robot movements, for example when the robot arms are moving to an appropriate pose before initiating edge tracing. During preprogrammed movements, arm-cloth collisions should be avoided, as they can undo all the previous unfolding steps in the pipeline. However, finding trajectories that avoid arm-cloth collisions is difficult when the cloth takes up a large volume in the already constrained workspace of the UR3e arms. This problem is amplified with stiff cloth, like the Beige sample, which maintains very wide horizontal dimensions when lifted, as opposed to flexible cloth that immediately sags downward. This leads to both a low edge tracing success rate and additional failures. The low edge tracing success rate is thus not inherent to the principles behind our methodology, rather its implementation on the UR3e platform. Also, grasping points are spread much farther apart in the horizontal directions than for more flexible samples, leading to a large number of discarded trials due to planning errors. This means that, with better planning on the same physical setup, there may be a larger number of trials failing two recoveries than reported in Table II.

To summarise, we have implemented UnfoldIR using two

UR3e robots with a limited work space. Consequently, the system in its current incarnation performs well for flexible cloth, but is ill-suited towards stiff cloth. In general, the main failure modes are due to lowest point detection and scripting preprogrammed motions to perform grasps. These issues are readily mitigated with more advanced visual data processing and planning algorithms.

Overall, UnfoldIR has proven to be an elegant solution to the unfolding problem. Previous work has shown the effectiveness of vision-only approaches, though the state-of-the-art is moving away from in-air unfolding, constraining the work area. We have now shown the potential of a predominantly tactile system with as little constraints on the environment as possible. This demonstrates that tactile feedback can be an invaluable complement to visual unfolding pipelines, and as such allows for new systems that quickly and effectively unfold clothing.

REFERENCES

[1] J. Zhu et al., ”Challenges and Outlook in Robotic Manipulation of Deformable Objects,” in IEEE Robotics & Automation Magazine, vol. 29, no. 3, pp. 67-77, Sept. 2022.

[2] H. Yin, A. Varava, and Danica Kragic, ”Modeling, learning, perception, and control methods for deformable object manipulation,” in Science Robotics, vol. 6, no. 54, May 2021.

[3] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, and P. Abbeel, ”Bringing clothing into desired configurations with limited perception,” IEEE Int. Conf. Robot. Autom. (ICRA), pp. 3893-3900, May 2011.

[4] H. Ha, and S. Song, ”FlingBot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding,” in Proc. 5th Conf. on Robot Learning (CoRL), vol. 164, pp. 24-33, November 2022.

[5] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, ”Speed-Folding: Learning Efficient Bimanual Folding of Garments,” IEEE Int. Conf. Intell. Robots Syst. (IROS), October 2022.

[6] A. Doumanoglou, A. Kargakos, T. -K. Kim and S. Malassiotis, ”Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning,” IEEE Int. Conf. Robot. Autom. (ICRA), pp. 987-993, September 2014.

[7] A. Gabas, Y. Kita, and E. Yoshida, ”Dual edge classifier for robust cloth unfolding,” in ROBOMECH Journal, vol. 8, iss. 1, April 2021.

[8] Y. Li et al., ”Regrasping and unfolding of garments using predictive thin shell modeling,” IEEE Int. Conf. Robot. Autom. (ICRA), pp. 1382-1388, May 2015.

[9] S.A. Wang, A. Albini, P. Maiolino, F. Mastrogiovanni, G. Cannata, ”Fabric Classification Using a Finger-Shaped Tactile Sensor via Robotic Sliding,” in Front. Neurorobot., vol. 16, February 2022.

[10] S. Tirumala, T. Weng, D. Seita, O. Kroemer, Z. Temel, and D. Held, ”Learning to Singulate Layers of Cloth using Tactile Feedback,” in IEEE Int. Conf. Intell. Robots Syst. (IROS), October 2022.

[11] Y. She, S. Wang, N. Sunil, A. Rodriguez, and E. Adelson, ”Cable manipulation with a tactile-reactive gripper,” in Int. J. Robot. Res. (IJRR), vol. 40, no. 12-14, pp. 1385-1401, August 2021.

[12] A. Billard, and D. Kragic, ”Trends and challenges in robot manipulation,” in Science, vol. 364, no. 6446, pp. eaat8414, June 2019.

[13] A. Verleysen, M. Biondina, and F. wyffels, ”Video dataset of human demonstrations of folding clothing for robotic folding,” in Int. J. Robot. Res. (IJRR), vol. 39, iss. 9, pp. 1031-1036, July 2020.

[14] T. Lips, V.-L. De Gusseme, and F. wyffels, “Learning keypoints from synthetic data for robotic cloth folding,” IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on Representing and Manipulating Deformable Objects, 2022.

[15] V.-L. De Gusseme and F. wyffels, “Effective cloth folding trajectories in simulation with only two parameters,” in Front. Neurorobot., vol. 16, 2022.