Reinforcement Learning for Robot-Assisted Dressing of People with Diverse Poses and Garments

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Bharath Hegde
ID No. 2020A7TS0143G

Under the supervision of:

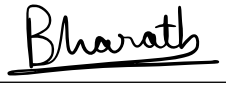
Prof. David Held, Prof. Zackory Erickson
&
Prof. Sougata Sen

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS
May 2024

Declaration of Authorship

I, Bharath Hegde, declare that this Undergraduate Thesis titled, 'Reinforcement Learning for Robot-Assisted Dressing of People with Diverse Poses and Garments' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 15 May 2024

Certificate

This is to certify that the thesis entitled, "*Reinforcement Learning for Robot-Assisted Dressing of People with Diverse Poses and Garments*" and submitted by Bharath Hegde ID No. 2020A7TS0143G in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Date: 15 May 2024

Supervisor
Prof. David Held
Associate Professor,
Carnegie Mellon Univerity

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS

Abstract

Bachelor of Engineering (Computer Science)

Reinforcement Learning for Robot-Assisted Dressing of People with Diverse Poses and Garments

by Bharath Hegde

Assistive robots have the potential to meet the increasing demand for home healthcare, particularly in crucial tasks such as dressing. However, robot-assisted dressing poses many challenges due to the deformable nature of clothing, variability in garments and human body shapes, and occlusions in the scene. Prior work has addressed some of these challenges but assumes a static arm pose during dressing and requires unoccluded observations. In this work, we investigate the performance of point cloud-based assistive dressing policies with partially observable information. We evaluate the effect of arm occlusions, the significance of arm points for the dressing task, and the impact of camera view angle on dressing performance. Our results show that policies trained with occluded observations perform similarly to those trained with unoccluded observations, that policy distillation is useful for generalisation to diverse regions and poses, even with occluded observations, and that the arm point cloud is significant for successful dressing. Furthermore, we find that the front camera view is better suited for the dressing task compared to the back view. Finally, we profile the control flow of real-world robot dressing to identify bottlenecks in dressing speed and propose improvements to increase the dressing frequency.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisors, Prof. David Held and Prof. Zackory Erickson, for their invaluable guidance throughout the course of this thesis. Their insights and advice has been instrumental in shaping the direction of my research and improving the quality of this work.

I am deeply thankful to Yufei Wang for his mentorship and guidance at every step of this endeavour. This thesis would not be possible without his patience and support.

I thank Carnegie Mellon University for providing a conducive and stimulating environment facilitating productive research. I extend my appreciation to BITS Pilani for making it possible to obtain first-hand experience of research in renowned institutes during my undergraduate degree. Last but not least, I am grateful to my on-campus supervisor, Prof. Sougata Sen, for his constant support and guidance.

Contents

Chapter 1

Introduction

With advancements in medicine and technology, more people are living longer than ever before. The high dependence of the ageing population on home-healthcare services, has caused a rise in demand of home healthcare [1]. Dressing is an example of one such important task, where almost 92 percent nursing and at-home care patients require assistance with dressing [4]. Assistive robots in home environments could potentially play an important role in meeting this demand, and relieve the need of human labour in these scenarios.

Dressing can be described as the manipulation of a clothing item in an accurate and safe manner over a human. In spite of the potential of assistive-dressing robots in this area, various aspects of this task pose challenges for robotics, perception and human interaction. The primary obstacle is the deformable nature of clothing items. Deformability results in complex movement and interaction dynamics which makes it difficult for a robot to predict its behaviour. The cloth can fold and wrinkle, causing occlusions that result in the partial observability of the state. It is also not simple to represent the state of a cloth mathematically like that of a rigid body.

The variability in the components involved in the task also pose a difficult challenge. Clothing items come in a multitude of shapes, sizes and material properties like texture and flexibility. Additionally, human bodies are also variable in terms of body shape and arm poses while dressing. A useful dressing robot should generalise and adapt well to the different scenarios. Finally, safety of the user being dressed is also an important concern. The robot must apply sufficient force to manipulate the cloth, but be gentle enough to the user while doing so. It should also learn to handle the real-time movements of the user and their reactions while dressing.

Prior works have attempted at tackling robot-assisted dressing by making simplifying assumptions, such as a single pose or a single garment [2], thereby having limited applicability to real dressing scenarios. Prior work [7] improves on these works through a point-cloud based reinforcement learning policy that is able to dress a diverse set of garments on a variety of human arm

poses through policy distillation. However, although this work is able to generalise to different garments and arm poses, it assumes a fixed or static pose during the course of dressing. With this assumption, the same arm pointcloud obtained before the cloth occludes the arm, is used throughout the dressing task.

A practical assistive dressing robot should be robust with respect to arm movements. In this case, the arm no longer has a fixed pose and a fixed arm pointcloud throughout the dressing task can no longer be assumed. Instead, the robot would have to make decisions based on partial information of the occluded arm and adapt it's actions to arm movements in real-time. In this work, we are motivated by this gap, and explore the following in relation to point cloud-based assistive dressing policies:

- Investigate the effect of arm occlusions on the dressing performance.
- Investigate the importance of the the arm points for the dressing task.
- Understand the effect of the camera view angle on the dressing performance.
- Profile the control flow to identify bottlenecks in the speed of real-world dressing.

Chapter 2

Background

2.1 Reinforcement Learning Setup

The dressing task is setup as a partially observable markov decision process and policies are trained using reinforcement learning in simulation. A brief summary of the components is defined below, and more details can be found in the original work [7].

- **Observation Space O:** Point clouds are used to represent the state of cloth, since it is not easy to represent the state of deformable objects compactly and point clouds offer a high dimensional state representation. The policy observation used as input is then a combination of the arm pointcloud, the garment pointcloud and the gripper point, which is the robot end-effector.
- **Action Space A:** The action is the delta transformation of the end-effector, represented by a 6D vector $\mathbf{a} = [\Delta x, \Delta y, \Delta z, \Delta\theta, \Delta\phi, 0]$, where $[\Delta x, \Delta y, \Delta z]$ denotes the delta translation of the end-effector in 3D space, and $[\Delta\theta, \Delta\phi, 0]$ represents the delta rotation of the end-effector using axis-angle representation, where roll rotation is set to 0.
- **Reward R:** The main component of reward is the task progression metric. The garment sleeve is approximated as a polygon, while the arm is represented as an intersection of two lines at the elbow. The distance of the intersection point between the polygon plane and the arm lines along the arm is used to quantify task progression. This is visualised in Figure 2.1. Additionally, there is also a force penalty to penalise excessive force on the user and a deviation penalty to prevent the robot from taking the cloth too far away from the user.
- **RL Algorithm:** The policy and Q function are represented by PointNet++ networks and Soft Actor Critic [3] is used as the RL algorithm. The policy architecture used is Dense

Transformation, where a segmentation-type PointNet++ outputs a per-point action vector as output, and the action executed is the action vector that corresponds to the gripper point.

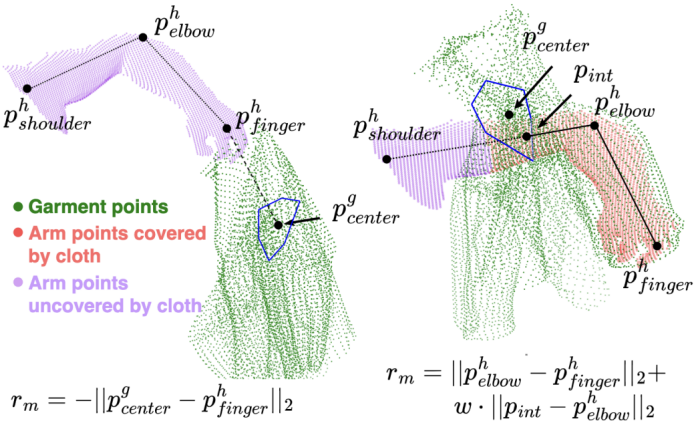


FIGURE 2.1: Task progression reward from [7]. Visualises the reward for dressing progress in the lower (Left) and upper (Right) regions of the arm.

2.2 Policy Distillation

The different possible arm poses, characterised by combination of arm joint angles, are divided into 27 regions. An expert policy is trained in simulation over each of the 27 regions. In order for a single policy to generalise over all the regions, policy distillation [6] is used where a student policy is trained in simulation over the entire 27 regions, while being guided by the expert teachers. This is done via the loss function, which is composed of 2 terms, the SAC loss and the distillation loss:

$$L(\theta_s) = L_{SAC}(\theta_s) + \beta \sum_{i=1}^{N} L_{distill}(\theta_s, \theta_t^i) \tag{2.1}$$

Here θ_s is the parameters for the student policy, θ_t^i is the parameters for the i^{th} student policy, β is the weighing parameter, $N = 27$ and the distillation loss is the earth mover's distance between the action distribution outputs of the student and teachers:

$$L_{distill}(\theta_s, \theta_t^i) = \sum_{n=1}^{B} \left(\mu_{\theta_s}(o'_n) - \mu_{\theta_t^i}(o_n)\right)^2 + \left(\sqrt{\sigma_{\theta_s}(o'_n)} - \sqrt{\sigma_{\theta_t^i}(o_n)}\right)^2 \tag{2.2}$$

In prior work [7] where the arm is assumed to be static and the complete unoccluded point cloud is used, $o'_n = o_n$, i.e the input to the student and the teacher, which is trained on unoccluded observations, are the same unoccluded observations o_n . In this work, we use occluded pointcloud observations o'_n as input to the student, while the expert teachers receive the unoccluded observations o_n, discussed in more detail in Chapter 3.

2.3 Simulator

Popular simulators mostly support only rigid-body dynamics and are thus not useful for the dressing task which requires deformable object simulation. The policies are thus trained in Softgym [5] which is based on the NVIDIA Flex simulator. Here, a cloth is simulated as a collection of particles connected by springs. The image-based observation space of Softgym is used, where an RGB image of the rendered environment is obtained and used by our our agent after conversion to a pointcloud. Softgym also provides pickers which are simplified versions of robot grippers modelled as spheres. These are useful for cloth manipulation since they can attach or detach to particles on the cloth to simulate gripping. Overall this simulator not only provides an interface supporting cloth manipulation, it also abstracts away the low-level details so that the focus can be on high level planning.

Chapter 3

Problem Statement

Although prior work [7] generalises well to dressing different garments over diverse arm poses, it assumes that the arm does not move during the course of dressing. With this assumption, the arm pointcloud is procured before the dressing begins (before the cloth starts to occlude the arm) and it is used as part of the input observation throughout the dressing task. In other words, the policy receives the complete or unoccluded pointcloud observation of the scene during both training and evaluation.

As discussed briefly in Chapter 1, in this work we are motivated by the notion that a general purpose dressing-robot must be robust with respect to arm movements and reactions of the user. It must be able to utilise the partial information in the scene that remains after occlusion of the arm by cloth. In this case, we will have to relax the assumption of having access to the complete or unoccluded pointcloud throughout the dressing task.

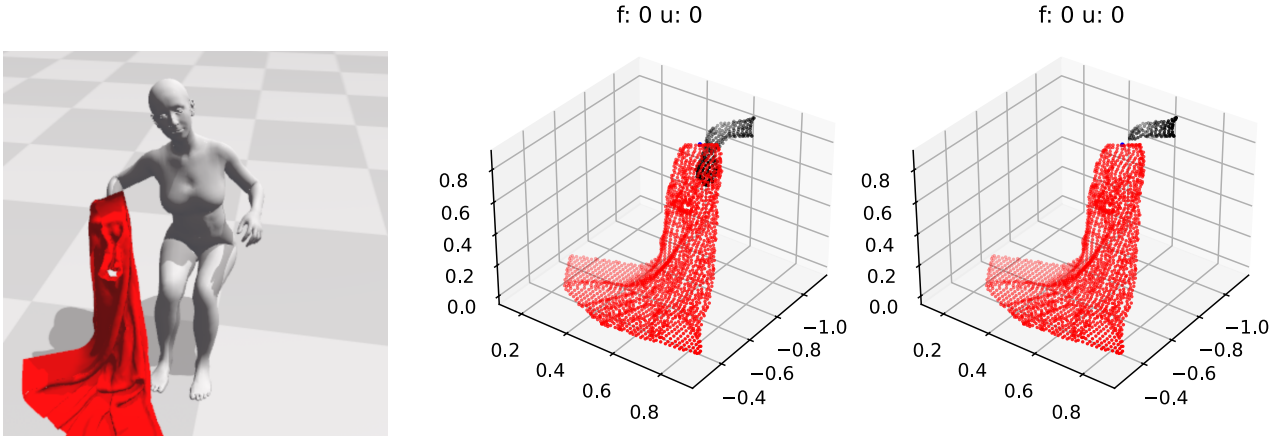
Instead, we would like to use the occluded pointcloud, obtained in real-time, to learn and inference about the optimal dressing method. To this end, we study the effect on dressing performance of different types of input observations with partial information - occluded, cloth-only and back-view camera observations, described in the following sections.

Assuming the policy learns to make optimal decisions based on partial information, the robot end effector movements still need to be fast enough to react and carry out the required actions. We thus profile the real-world dressing control flow to identify bottlenecks in the action execution, results of which can be found in section 4.4.

3.1 Occluded Observation

In the real world, the real-time pointcloud received from the camera is by itself the occluded observation. In simulation, we would like to visualise the arm and cloth pointclouds separately.

We do this by first rendering the arm and cloth separately, obtaining their depth images and then discarding those depth pixels in the arm that are farther away from the camera than the cloth and vice versa. By retaining only those depth pixels closest to the camera, we obtain the partially visible or occluded pointcloud, which is the actual observation seen by the camera. Figure 3.1 compares an example of unoccluded observations that prior work [7] uses, with occluded observations that we wish to analyse the performance of the RL policy with.



(a) (b) (c)

FIGURE 3.1: Visualisation of occluded and unoccluded observations [1]: (a) The camera view in simulation, (b) Unoccluded pointcloud observation, (c) Occluded pointcloud observation

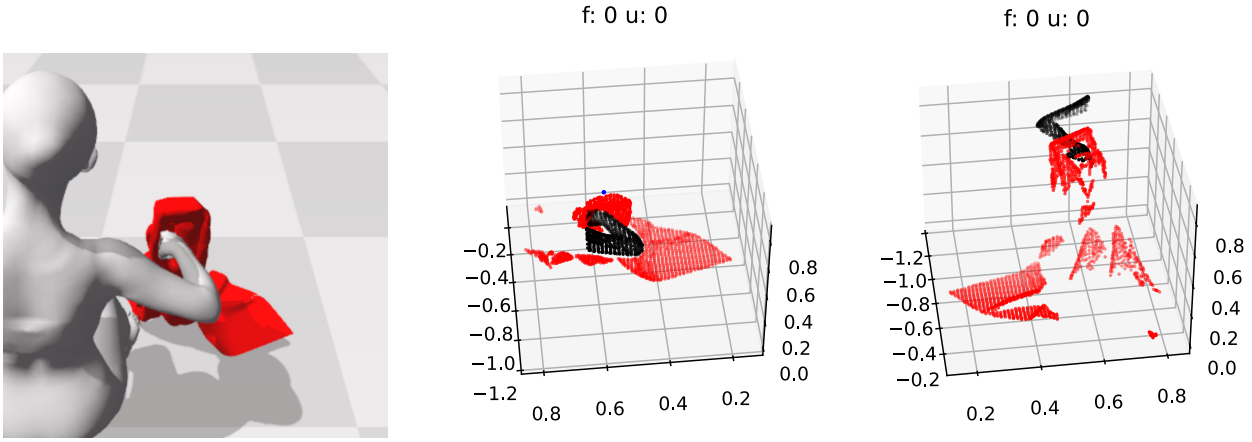
3.2 Cloth-only Observation

To better understand the effect of arm occlusions, we also train policies in simulation in the extreme case of using just the cloth poincloud observations, without any arm pointcloud. This would correspond to the pointcloud in Figure 3.1c, but with only the red portion of the observation retained.

3.3 Back-view Camera Observation

Occlusions are heavily dependent on the viewing angle; information visible in one camera angle would be different from another. For example, with a front view camera, varying lengths of the arm is occluded during dressing. On the other hand, with a back view camera angle, since the arm is closer to the camera than the cloth, the arm is more visible throughout dressing when compared to the front view angle. So it would be useful to know the effect of the viewing angle on policy performance.

[1] Downsampling decreased for better understanding



(a) (b) (c)

FIGURE 3.2: Visualisation of back-view camera observation: (a) The back camera view in simulation, (b) Corresponding pointcloud observation, (c) Same pointcloud in a different orientation

The above figure shows an example of a back-view camera occluded pointcloud observation. As seen in 3.2c, a larger portion of the arm is visible when compared to the front view observation in 3.1c. However, in the back view, the arm appears in front of the cloth. This causes occlusion of the the cloth as can be seen by the gaps of the red pointcloud in 3.2c.

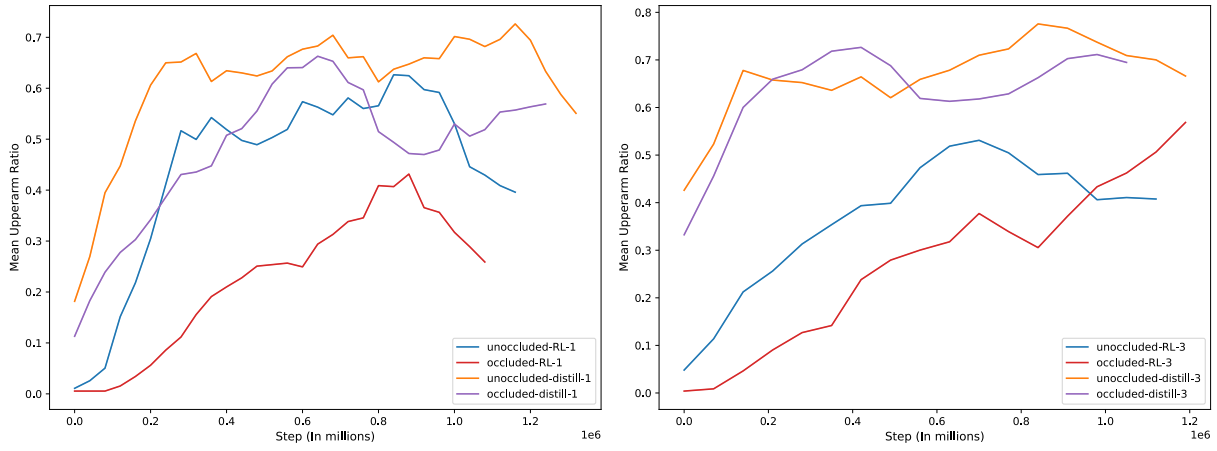
3.4 Experimental Setup

As in work [7], arm poses are divided into 27 regions based on shoulder, inwards-outwards elbow and upwards-downwards elbow joint angles. Each region contains 50 different arm poses. Out of these, 45 are used for training and 5 are used for evaluation. The garments used in training are a hospital gown and 4 cardigans, which vary in shapes and sizes. During each episode of training a random garment (out of 5), region (out of 1, 3 or 27 depending on the regions being trained on) and pose (out of 45) is chosen. During evaluation, every garment and region is evaluated on the 5 remaining poses of that region. A metric 'mean_upperarm_ratio' is calculated during evaluation which is the ratio between the dressed upper arm distance and the actual upper arm length. Here upperarm refers to the region between the elbow and the shoulder, and this metric gives an indication of how far along the arm the policy was successful in dressing the cloth on average across the regions and garments. In this work, in order to speed up evaluation time for the 27 regions case, we evaluate 2 random garments on a single pose for each of the 27 regions.

Chapter 4

Experiments and Results

4.1 Occluded Observation



(a) (b)

FIGURE 4.1: Dressing performance over (a) 1 region (b) 3 regions

For better interpretability, we first evaluate dressing performance over 1 and 3 regions in 4.1 and then over 27 regions in 4.2. In each case, we plot policies trained using RL directly over the regions and student policies distilled from the expert teachers corresponding to the regions, for both occluded and unoccluded observations.

From the plots, it is clear that disitilled policies outperform vanilla RL ones, in both occluded and unoccluded cases. The gap between distillation and RL increases as we move from 1 region to 27 regions. This aligns with our expectation since it is easier to obtain success in a single region by learning to manipulate the cloth in a general direction while dressing. However, with more regions this is no longer feasible and the policy needs to utilise the pointcloud information better in order to learn dressing over a wide range of poses.

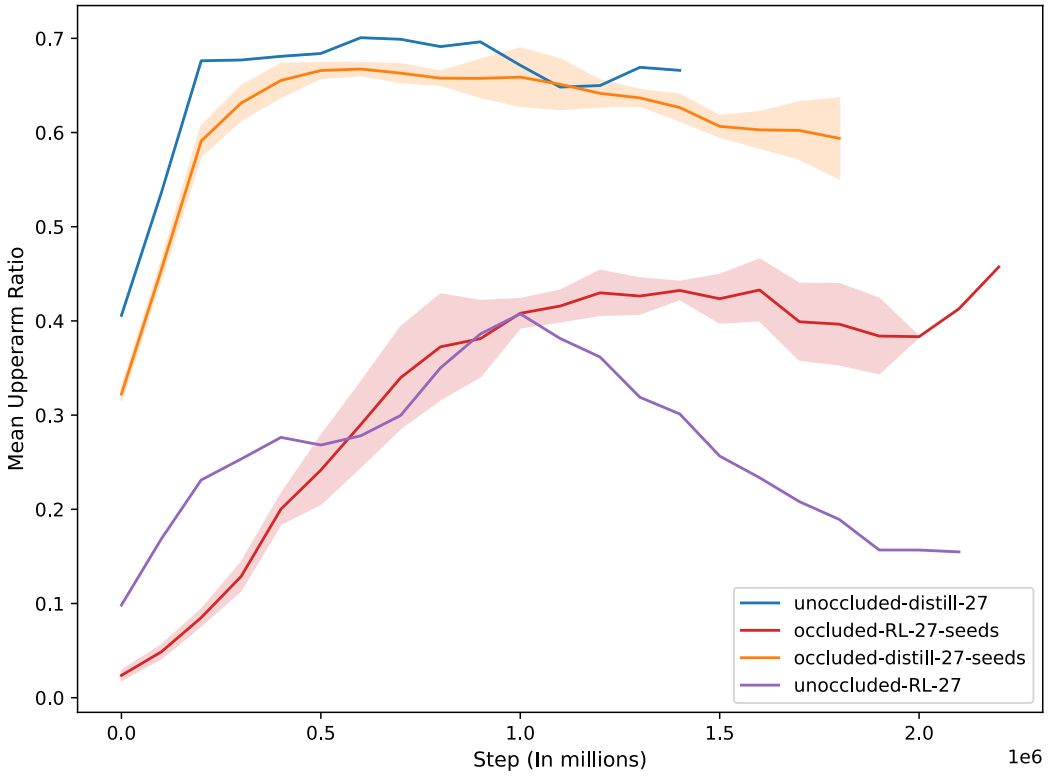


FIGURE 4.2: Dressing performance over 27 regions [1]

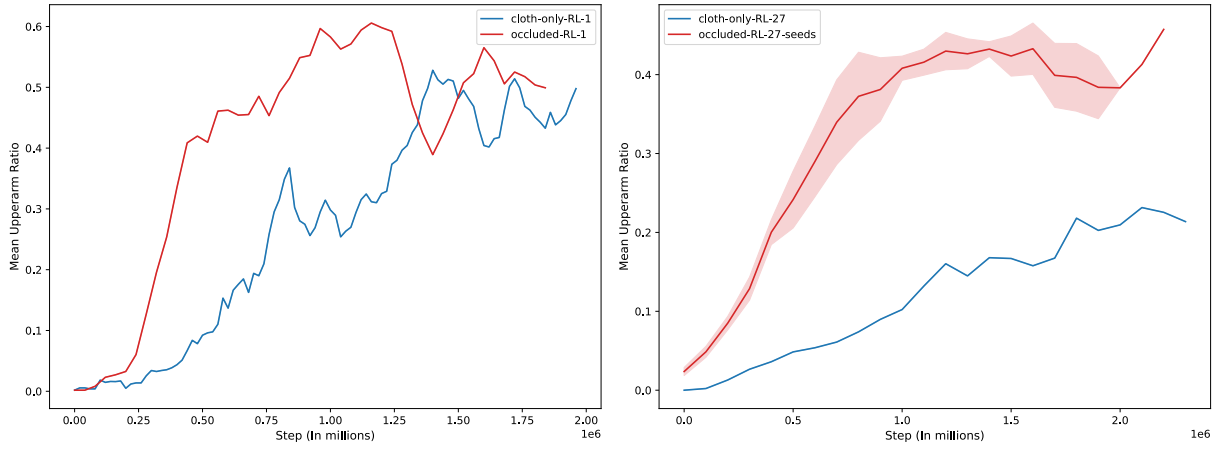
When comparing the occluded polices with the unoccluded, we notice an interesting result. One might expect the occluded policies to perform worse since they lack information of portion of arm hidden by the cloth. However, from the plots we observe that policies trained with occluded observations perform similarly with the ones trained with unoccluded observations. In fact, in 4.2, the occluded RL policy performs slightly better than the unoccluded one. We can draw two possible conclusions from this; one, that the history or occluded portion of the arm itself is not significant to complete the dressing, rather the undressed portion of the arm is more important, or two, that cloth pointcloud by itself carries sufficient information for the dressing, so the type of arm pointcloud does not affect the performance much. To understand this better we analyse performance using just cloth pointcloud observations in the next section.

4.2 Cloth-only Observation

Looking at 4.3a the policy trained with only the cloth observations, reaches similar performance as the policy trained with both arm and cloth observations. However, as discussed in the previous section, it is easier for an otherwise bad policy to perform well, when evaluated on a single region. Looking at 4.3b, the cloth-only policy is clearly outperformed by the occluded policy. We can

[1] Shaded regions are the average over multiple seeds of the same policy

now conclude that the arm pointcloud is in fact significant for dressing, and rule out the second possibility in the previous section.



(a) (b)

FIGURE 4.3: Cloth-only dressing performance over (a) 1 region (b) 27 regions

4.3 Back-view Camera Observation

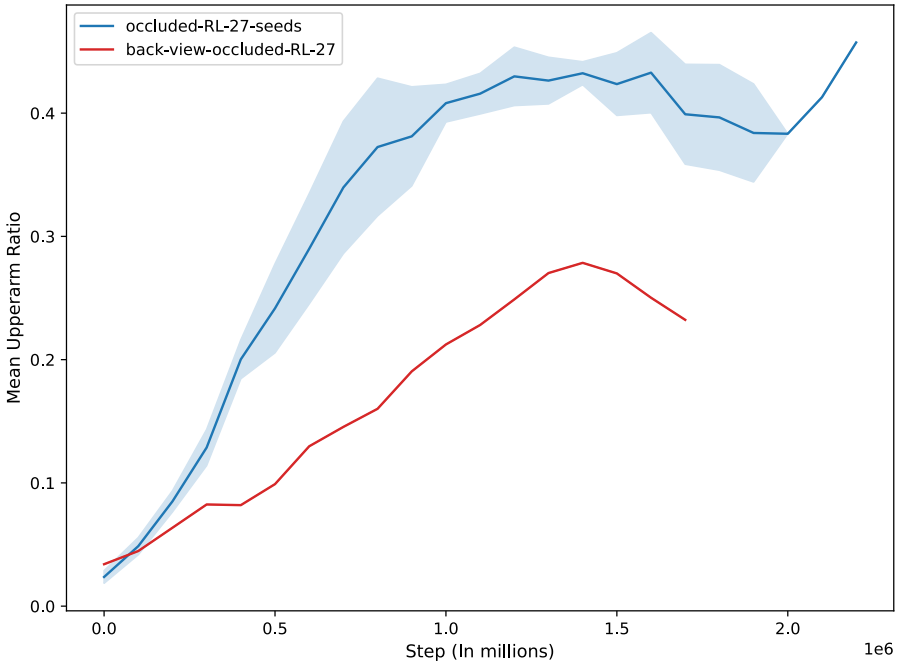


FIGURE 4.4: Back view dressing performance over 27 regions

The policy trained using front view observations (red) performs better than the one trained using back view observations (red). Although, more of the arm is visible in the back view, much of the cloth is occluded too, as seen in 3.2c. We conclude that one, the angle of view during the

dressing task is signficant for dressing performance and two, the front view is a better choice for dressing. However, one approach that can be tried in the future is combining both views to incorporate information of the arm and cloth from both views.

4.4 Benchmarking of Dressing Control Flow

In simulation, the policy learns intelligent behaviour to tackle some key aspects of the dressing task. One example is during insertion of arm into opening sleeve of the garment, where this a high chance of failure due to the cloth getting stuck at the fingers. The policy learns to 'shake' the cloth when close to the fingers, thereby increasing the chance of the arm entering the sleeve. Another such behaviour is the policy learns to retrack, undress and re-dress in situations where the cloth gets stuck. These examples can be viewed here and here. Such behaviour cannot be reproduced in the real world, unless the robot is not limited in its ability to execute actions fast. Additionally, for the robot to be able to react to movements in the arm, it needs to rapidly react to changes in the input observations.

There is thus a need to improve the current dressing frequency of 1.09Hz, representing 1.09 actions taken per second. We do this by profiling the dressing control flow with respect to time and identifying the bottlenecks in the dressing speed.

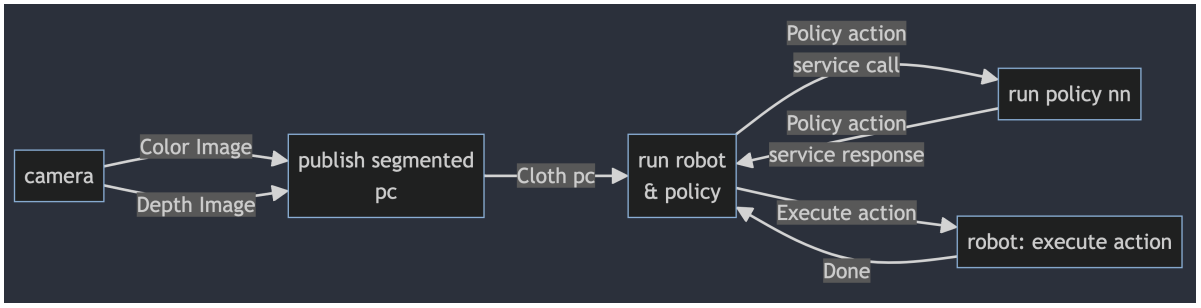


FIGURE 4.5: Control flow of real world robot dressing

The control graph of the real world robot dressing is illustrated in 4.5. The camera continually publishes synchronised colour and depth images, which is subscribed to by the 'publish segmented pc' node, which in turn publishes the pointcloud observation, after segmenting out the relevant section of the scene. 'run policy nn' loads the trained policy and provides a ROS service to compute the next action given the policy observation. 'run robot & policy' is the main node for handling the dressing task. It subscribes to the published pointcloud observation, calls the 'run policy nn' action service and receives the next action from it, performs motion planning and sends commands to the robot for execution in the real world.

Profiling was done by averaging the time taken for the various functions to execute over multiple runs of the dressing task. These results are compiled in 4.1.

Script	Average time per call (in seconds per cycle)
(a) Robot: execute action	0.44
(b) Run policy nn (Action service call)	0.21
(c) Receive published segmented pc	0.18
(d) Other	0.08
Total	0.91

Table 4.1: Average time per funciton call

The current average time per action cycle is 0.91s, or 1.09 actions per second. From the above table, it is clear that the largest bottleneck in dressing is the actual action execution on the robot. The speed of execution depends on the robot itself and cannot directly be improved. However, we can try and decouple our control flow from the robot's action execution: currently the control flow is serially executed, i.e all the calls are blocking, and the 'run robot & policy' node waits until the robot completes its action execution. We can modify this to a non-blocking call, so that while the robot executes the current action, the next pointcloud is obtained and sent to the policy for the next action. This way, we can decrease the the average cycle time to 0.44s, improving the dressing frequency almost two times to 2.2 actions per second. Additionally, since the action service call to the policy also takes significant time (0.21s), this can be combined with the 'run robot & policy' node for more efficiency.

Chapter 5

Conclusions and Future work

In this work, we investigated the performance of point cloud-based assistive dressing policies with partially observable information. Our experiments demonstrate that policies trained with occluded observations perform similarly to those trained with unoccluded observations, indicating that the undressed portion of the arm is more significant for successful dressing than the occluded portion. We also found that the arm point cloud itself is crucial for the dressing task, as policies trained with only cloth observations were outperformed by those trained with both arm and cloth observations. We also showed that the camera view angle affects dressing performance, with the front view being better suited for the task compared to the back view. Lastly, by profiling the control flow of real-world robot dressing, we identified bottlenecks in dressing speed and proposed improvements to help increase the dressing frequency.

Going forward, further efforts are needed to transfer our occluded policy in simulation to the real word. While our work showed that the front camera view is better for dressing compared to the back view, combining information of the arm and cloth from both views can be explored to decrease occlusions and potentially improve dressing performance. Additionally, although our work addresses arm occlusions, further research is needed to develop policies that can dynamically handle arm movements during dressing.

Bibliography

[1] *Current Home Health Care Patients*. 2004. URL: https://www.cdc.gov/nchs/data/nhhcsd/curhomecare00.pdf.

[2] Zackory Erickson et al. *Multidimensional Capacitive Sensing for Robot-Assisted Dressing and Bathing*. 2019. arXiv: 1904.02111 [cs.RO].

[3] Tuomas Haarnoja et al. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. arXiv: 1801.01290 [cs.LG].

[4] Lauren D. Harris-Kojetin et al. *Long-term care providers and services users in the United States, 2015-2016*. eng. Journal Issue. Hyattsville, MD, February 2019. URL: https://stacks.cdc.gov/view/cdc/76253.

[5] Xingyu Lin et al. *SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation*. 2021. arXiv: 2011.07215 [cs.RO].

[6] Andrei A. Rusu et al. *Policy Distillation*. 2016. arXiv: 1511.06295 [cs.LG].

[7] Yufei Wang et al. *One Policy to Dress Them All: Learning to Dress People with Diverse Poses and Garments*. 2023. arXiv: 2306.12372 [cs.RO].