2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Soft Tissue Simulation Environment to Learn Manipulation Tasks in Autonomous Robotic Surgery*

Eleonora Tagliabue[1†], Ameya Pore[1†], Diego Dall'Alba[1], Enrico Magnabosco[1], Marco Piccinelli[1] and Paolo Fiorini[1]

***Abstract*— Reinforcement Learning (RL) methods have demonstrated promising results for the automation of subtasks in surgical robotic systems. Since many trial and error attempts are required to learn the optimal control policy, RL agent training can be performed in simulation and the learned behavior can be then deployed in real environments. In this work, we introduce an open-source simulation environment providing support for position based dynamics soft bodies simulation and state-of-the-art RL methods. We demonstrate the capabilities of the proposed framework by training an RL agent based on Proximal Policy Optimization in fat tissue manipulation for tumor exposure during a nephrectomy procedure. Leveraging on a preliminary optimization of the simulation parameters, we show that our agent is able to learn the task on a virtual replica of the anatomical environment. The learned behavior is robust to changes in the initial end-effector position. Furthermore, we show that the learned policy can be directly deployed on the da Vinci Research Kit, which is able to execute the trajectories generated by the RL agent. The proposed simulation environment represents an essential component for the development of next-generation robotic systems, where the interaction with the deformable anatomical environment is involved.**

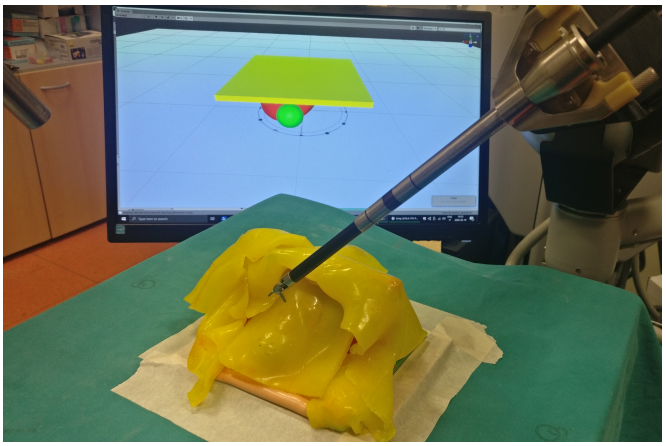


Fig. 1. In our setup, a single arm of the dVRK interacts with silicone fat tissue covering a kidney phantom. The simulated scene controlling dVRK movements can be seen in the background.

I. INTRODUCTION

Diffusion of Surgical Robotic Systems (SRSs) has constantly increased over the last 15 years. This spreading is motivated by enhanced dexterity and 3D vision provided by SRSs that allows to translate a conventionally difficult Minimally Invasive Surgical (MIS) procedure into an easier work for surgeons and, ultimately, to better surgical treatment for patients [1]. Since all MIS procedures are composed of a series of standard subtasks (e.g. dissection, suturing and knot tying), recent SRS research has pushed towards their robotic automation, to allow the surgeon to focus on more critical steps and further improve the overall quality of the surgical treatment. One of the most common subtasks present in most MIS procedures is soft tissue manipulation. For example, in robotic assisted nephrectomy procedures, the surgeon has to manipulate the highly deformable perirenal fat tissue which covers the kidney in order to expose, and subsequently have access to, the region of interest. The main challenge when attempting to automate robotic tissue manipulation relies on accounting for the dynamic behavior of soft tissues interacting with the anatomical environment. The design of hand-crafted control policies requires to consider such dynamic behavior in path planning and control methods. However, the high variability of soft tissue properties makes it very complex to find a control policy able to generalize to realistic anatomical environments.

Deep Reinforcement Learning (DRL) have shown promising results in the automation of robotic tasks, without the need to design ad-hoc control strategies [2]. However, the DRL agent reaches robust performance after it has explored a huge amount of possible policy options, which requires a long training consisting of a large number of attempts, many of which fail with unsafe behavior. Although safe learning methods have been proposed, their exploitation in the surgical robotic field remains limited, due to the impracticality to acquire many training trials from real SRSs [3]. To tackle this issue, some works have proposed to use a sim-to-real approach, where autonomous agents can be trained in a simulated environment and the learned policies can be successfully transferred to a real system [4], [5]. Training an agent in simulation seems the most appropriate strategy to apply DRL to learn surgical subtasks, since the training process can be optimized, avoiding all the limitations related to technical and ethical aspects of organizing clinical experiments. However, to exploit sim-to-real methods in the surgical robotic field, it is essential to have realistic and fast simulation environments where to prototype and test the algorithms. Realism is an important requirement of the simulation framework used, in fact it has been proved that

[1]Authors are with Department of Computer Science, University of Verona, Verona, Italy eleonora.tagliabue@univr.it

† Eleonora Tagliabue and Ameya Pore contributed equally to the paper.

*This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 742671 "ARS") and under the Marie Sklodowska-Curie (grant agreement No. 813782 "ATLAS").

the greater the discrepancy between the simulated and real environments (called *reality gap*), the higher the probability that a policy learned in simulation will perform poorly in the real world [6].

In this work, we introduce a completely general and modular framework that allows to exploit RL methods to learn task automation in simulated surgical environments which involve deformable objects. The main contributions of this work are the following:

1) we present a complete open-source[1] virtual environment designed for training RL methods involving deformable objects interaction;
2) we show that a soft tissue manipulation task can be learned in simulation without any user demonstration;
3) we demonstrate that the learned policy translates directly to the surgical robotic system thanks to the da Vinci Research Kit (dVRK), without further training.

To the best of our knowledge, this represents the fist attempt of introducing an extensible virtual environment supporting soft tissue simulation for training RL agents in surgical robotic applications. By making it publicly available, we hope to encourage the research community to adopt it for modelling many different scenarios of increased complexity. Furthermore, the simulation framework is designed to enable a straightforward deployment of the learnt behaviours on real SRSs.

II. RELATED WORKS

Automation of surgical subtasks with the dVRK is a very active research area within the surgical robotics community. Most of the prior works in the field rely on the Learning From Demonstrations (LfD) approach, where the robot learns to perform a task in complete autonomy from a set of demonstrations from expert users [7]–[9]. LfD represents the preferred approach since it does not require to explicitly model the highly deformable anatomical environments the robot interacts with, which would be the most challenging step. However, the robustness of learned tasks to changes in the initial conditions or in the environment is strongly affected by the amount and variety of expert demonstrations provided to the system. Collecting such a dataset is impractical and often unfeasible in clinical settings. Some alternative approaches for surgical actions automation which exploit simple control algorithms have also been proposed, but they require the extraction of visual and geometric features that would need strictly controlled conditions and are thus difficult to generalize in realistic settings [10]. Leveraging on the latest advancements in DRL, which has proved to be promising for learning control strategies in robotics when the behavior of the environment is not completely known, some recent works have focused on applying DRL methods to learn tasks in robotic surgery [10], [11]. In their recent work, Richter et al. proposed the first RL environment for surgical robotics which allows to train DRL policies in simulation before moving to the real SRS, where OpenAI Gym communicates with V-REP robotic simulator [5]. The main limitation of such framework is that it lacks the possibility to model deformable objects, which is extremely important in a realistic simulation, especially if the agent has to interact with the anatomical environment.

In general, there are very few works of sim-to-real RL which involve the manipulation of deformable objects, even fewer if we consider the surgical field [12]. Shin et at. have compared the performances of RL and LfD in automating soft tissue manipulation with the Raven IV surgical system [13]. Although both RL and LfD could successfully learn the task, trajectories learned with RL have not been transferred to the real robotic system because considered too hazardous. If an integrated framework able to provide a more realistic simulation of the environment dynamics were available, it would lead to different conclusions. Surgical pattern cutting learned with DRL in simulation involving deformable objects has been successfully tested on the dVRK in [4], [11]. Authors have used an ad-hoc simulation environment for RL policy learning, which can only simulate a 2D deformable sheet with a set of masses and springs, and cannot thus generalize to more complex deformations and anatomical shapes.

III. ROBOT PLATFORM AND SIMULATION ENVIRONMENT

In this work, we show that an RL agent can be trained in our simulated environment to manipulate soft tissues and the learned policy can be deployed to the da Vinci Surgical System, controlled through the dVRK [14].

A. Robot Platform

We consider a scenario with a single dVRK slave unit, known as Patient Side Manipulator (PSM) arm. In order to have a state space which is straightforward to observe for RL in both the simulator and the real robot, the motion of the PSM end-effector (EE) is controlled in the Cartesian space, keeping the EE orientation constant (as in [5])). Although in this work we consider the Large Needle Driver (LND) instrument unit, our framework generalizes to any possible instrument, provided that the corresponding kinematic model is loaded in simulation. Therefore, the PSM EE state is completely characterized by its position $\mathbf{p}_t$ and gripper state ($g_t \in \{0, 1\}$, open/close). Similarly to [5], we normalize the PSM positions with respect to the workspace, defined by the PSM joint limits and the obstacles in the environment, to allow generalization of the learned policies to various joint configurations. We assume that the 3D model of the anatomical environment is available, extracted from some kind of pre-operative images such as Magnetic Resonance Imaging (MRI), which allows us to know the position of the tumor area of interest $\mathbf{q}$.

B. Simulation Environment

Our simulation framework is based on Unity3D engine, a game development platform which has shown promise in medical simulations [15]. The main advantage of using Unity

[1]Project available at https://gitlab.com/altairLab/unityflexml

is that its high modularity allows users to easily customize the environment scene and to exploit advanced features implemented in separate plugins. In particular, our framework relies on two main Unity plugins: the Machine Learning Agents Toolkit (ML-Agents), for training intelligent agents [16], and NVIDIA FleX, for soft object deformation [17].

Deformable bodies are simulated with the Position Based Dynamics (PBD) approach relying on the optimized implementation provided by NVIDIA FleX, which has already proved able to model soft tissue deformations both realistically and fast [15]. Therefore, it is suitable for performing the huge number of trial and error attempts required by RL. In an effort to minimize reality gap, we perform a preliminary optimization procedure to find the PBD deformation parameters that keep the real and the simulated behaviors as close as possible. From what concerns the simulation of the robotic part, we have implemented a closed form inverse kinematics of the PSM to enable the Cartesian space control of the manipulator. At each simulation step, the robotic system is allowed to perform a very small motion increment. Thanks to this feature, we could assume the impact of the robot dynamic behavior to be negligible, thus we have not accounted for it in the simulation [5]. In our simulated environment, grasping of an object is modelled as an atomic event triggered when the relative distance with the EE is less that $2\,mm$.

IV. LEARNING SOFT TISSUE MANIPULATION

In this work, methods of RL are employed to learn the actions that the da Vinci system has to perform to expose a kidney tumor by manipulating fat tissue.

A. Background in RL

Our RL problem is formulated as a Markov Decision Process (MDP) where an agent learns by interacting with the surrounding environment. An MDP is described as a tuple (S, A, r, P, γ), where S is the state space, A is the action space, $r : S \times A \rightarrow \mathbb{R}$ is the reward function that encapsulates the goal, $P : S \times A \times S : \mathbb{R}$ is the transition probability function, and $\gamma \in [0, 1]$ is the discount factor. At each time step t, the environment produces a state observation $s_t \in S$. Then, the agent takes an action $a_t \sim \mu(s_t)$, $a_t \in A$, sampled from $P(s_t, a_t, .)$, which brings it to a new state s_{t+1}. The goal of the agent is to learn the behavior policy $\mu : S \rightarrow A$ which allows to maximise the expected discounted reward $E[\sum_{i=0}^{T-1} \gamma^i r_i]$, where T is the considered time horizon.

B. Observation and Action Space

In our problem, the agent is represented by the EE of the da Vinci PSM, which interacts with the surrounding anatomical environment, whose initial state is assumed to be known from pre-operative data. Our task consists of moving the PSM arm from a pre-defined initial position $\mathbf{p}_0$ to a position close to the tumor $\mathbf{q}$, grasp the fat and lift it to a pre-defined final position $\mathbf{p}_T$. Points $\mathbf{p}_0$ and $\mathbf{p}_T$ are considered fixed through all the training experiments, to accelerate the training phase by reducing the number of possible configurations.

The state and action space of the environment is:

$$\begin{aligned} S_t &= [\mathbf{p}_t, \mathbf{q}, \mathbf{p}_T, ||\mathbf{p}_t - \mathbf{q}||, ||\mathbf{p}_t - \mathbf{p}_T||] \\ A_t &= [\mathbf{\Delta}_t, g_t] \end{aligned} \tag{1}$$

where $||.||$ is the Euclidean distance. $\Delta_{t,i} = 0.5\alpha$, $\alpha \in \{0, -1, +1\}$ tells the agent if it has to remain still, move backward or forward by $0.5\,mm$ in the i_{th} spatial dimension, while $g_t \in \{0, 1\}$ represents the gripper state (open/close).

C. Reward Function

We design a reward function which changes depending on the current gripper state:

$$r(s_t) = \begin{cases} ||\mathbf{p}_t - \mathbf{q}|| * k - 0.5, & \text{if } g_t = 0 \\ ||\mathbf{p}_t - \mathbf{p}_T|| * k, & \text{if } g_t = 1 \end{cases} \tag{2}$$

where k is a normalization factor which depends on the volume in which PSM can move. When the gripper is open, the reward encourages the PSM to move towards the tumor. On the other hand, when the EE has grasped the tissue, it is pushed towards the target position. During training, the rewards are accumulated at each episode. An episode terminates after 1500 steps. Our DRL framework is based on Proximal Policy Optimisation (PPO), in its implementation provided by Unity MLAgents.

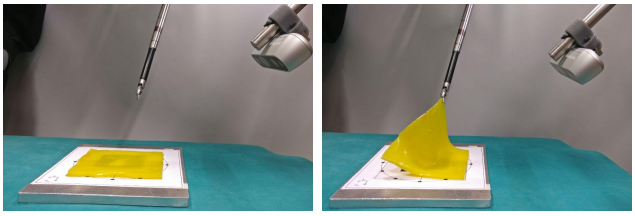
V. EXPERIMENTS

The soft tissue manipulation task we aim to learn consists in grasping and pulling the fat tissue covering the kidney in order to expose a tumor. Our real experimental setup consists of a synthetic kidney phantom covered with silicone fat tissue, shown in Fig. 1. We restrict the portion of fat tissue our agent interacts with to a 90x$90\,mm$ square region which is rigidly anchored to the top part of the kidney. The silicone patch representing the fat is held in place through a custom designed rigid structure, which enables to uniquely define the position of fat and kidney between the simulated and the real environment. All the simulation experiments, including RL training and dVRK control, ran on a workstation equipped with an AMD Ryzen 3700X processor and NVIDIA TitanX GPU.

A. Simulation Optimization

To ensure that the simulation used to train our RL agent provides a realistic behavior, we employ the genetic algorithm scheme to optimize PBD parameters most impacting the deformable behavior of the fat tissue our robot interacts with, similarly to [15]. Optimization is performed on some preliminary experiments where a teleoperated PSM arm lifts the fat tissue, which starts from a planar configuration and is rigidly fixed on one side (Fig. 2). We define $N = 5$ different pinch points along fat contour and $L = 3$ different levels of lifting are defined for each pinch point. The point cloud representing ground truth positions of the fat tissue is acquired using an Intel RealSense D435 Depth camera (Intel Corporation, Santa Clara, USA), whose position is defined

with respect to a custom calibration board which allows to rigidly align the simulated and the real environment (Fig. 3).



(a) (b)

Fig. 2. One of the experiments of the optimization process. The fat tissue is anchored to the calibration board (right side in the Figure). (a) Rest condition; (b) Deformed condition. Point cloud of the deformed tissue is acquired with the depth camera shown on the right.

Optimal values for the cluster spacing, cluster radius and cluster stiffness parameters (those mostly controlling the PBD implementation of NVIDIA FleX) are estimated by minimizing the following error ϵ:

$$\epsilon = \frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{m=1}^{M} ||\mathbf{x_{PBD}}(l,n) - \mathbf{x_{PCL}}(l,n)|| \qquad (3)$$

where $||.||$ represents the Euclidean distance between the position of the M particles defining the fat in simulation $\mathbf{x_{PBD}}$, at deformation level l and pinch point n, and the closest point of the corresponding point cloud $\mathbf{x_{PCL}}$. The acquired point cloud has been decimated to bring the number of points comparable to M. The diameter of the PBD particles is set to $3\,mm$ (i.e., the width of our tissue sample), which allows to describe the dynamics of the fat tissue with a single layer of particles. The constraints and the range of allowed values for each parameter are set according to [15].

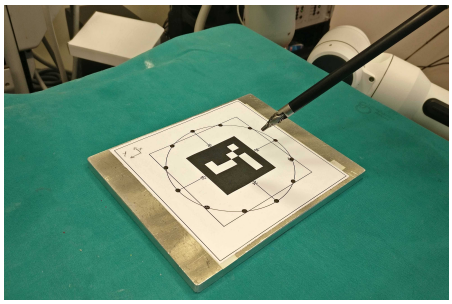


Fig. 3. The calibration board used to uniquely map all the components of our experimental setup to the simulated environment.

B. Simulation

In our pipeline, the agent is trained in simulation always keeping the same initial and final positions ($\mathbf{p}_0$ and $\mathbf{p}_T$). In order to assess if the behavior learned by the agent is robust to different starting EE positions, we perform an experiment where the trained agent has to perform the task starting from 100 different positions uniformly sampled above the portion of the fat tissue of interest. The task is considered successful if, once the robot EE reaches the final target position, the tumor region becomes visible. Tumor exposure is assessed as percentage of tumor surface which can be seen from a simulated endoscope placed in front of the kidney in the simulation scene (the camera point of view can be appreciated in Fig. 6). To do that, we count the number of pixels belonging to the tumor (i.e., green pixels in our simulation) on the simulated camera image after each grasp. This evaluation allows us to assess if the quality of the exposure is correlated to the initial PSM position.

C. Robotic Setup

The most important preliminary step that needs to be performed in order to transfer the learned policy from the simulation scene to the real dVRK system consists in the rigid alignment between the real and the simulated environment. In our setup, we map the poses of the PSM in a common reference space by reaching several points on the calibration board shown in Fig. 3. Precise alignment of the two environments is extremely important for our application since all the movements of the da Vinci arm in the real system are directly controlled by the simulated environment. Furthermore, due to the fact that we do not rely on any visual feedback for these preliminary experiments, grasping event is triggered in simulation whenever a collision event between the end-effector and the fat is detected, and the corresponding action is transferred to the real system. Therefore, accurate registration is essential to prevent inconsistencies between the two environments. The policy transfer is evaluated by replicating the simulated scene with the PSM arm starting from 25 different initial positions equally distributed above the lower half of the fat tissue. The proximal part of the fat tissue has not been considered due to safety constraints (e.g., to avoid potential collisions between the robot and the rigid supporting structures). For each starting position, we evaluate the number of times the real system is able to successfully grasp the tissue. Data exchange between the simulated environment and the real one exploits UDP-based communication, as described in [18], at rate of $50\,Hz$.

VI. RESULTS

A. Simulation Optimization

Optimal values for the cluster spacing, radius and stiffness parameters generated with the optimization process are 0.127, 0.095 and 0.361 respectively, which lead to an average error between the simulated and the ground truth point clouds of approximately $3\,mm$, comparable with dimensions of PBD particles. These values are employed to describe the deformable behavior of the fat tissue in the simulation scene.

B. Simulation

Fig. 4 shows the obtained learning curve. The agent takes 3 million steps to learn the whole task. This result is comparable with the ones obtained from previous works based on rigid bodies simulation, for instance [5]. Analyzing the reward trend, it emerges that the agent needs approximately 500 thousand steps to learn the approach behavior towards

the fat, the following 1,5 million steps to learn the interaction with the fat and eventually the last 1 million steps to learn the retract behavior after grasping. Note that the behavior is learned without any visual cues and purely on the basis of positional information.

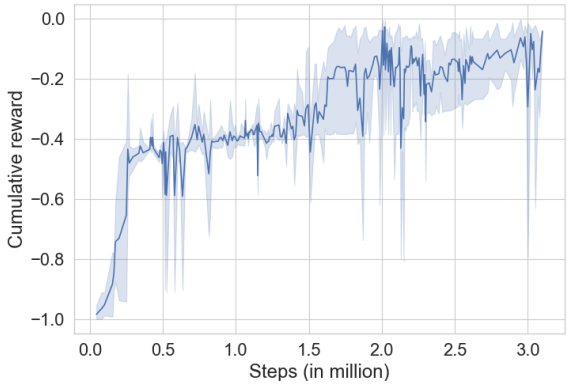


Fig. 4. The obtained learning curve. Cumulative reward is normalised in the range $[-1, 0]$. The shaded area spans the range of values obtained when training the agent starting from three different initialization seeds.

The plot in Fig. 5 shows the percentage of tumor surface which is visible from the simulated camera depending on the starting position of the PSM arm above the fat tissue. The starting position used for the training correspond to the point $(6, 5)$ of the evaluation grid. Whenever the agent starts from the distal part of the tissue (i.e., the one farther from the fixed region), the agent learns to grasp the tissue and the tumor becomes visible from the camera (at least partially). The learned policy is thus able to generalize to different initial EE positions, except when they are close to the proximal area of the kidney. This behavior can be further appreciated in Fig. 6, which shows simulation results corresponding to two different initialization points. It seems that, when $\mathbf{p}_0$ is initialized close to the fixed fat region, the agent is not able to move towards a reasonable grasping point, thus causing the tumor not to be exposed. This may be due to the fact that our reward function does not penalize the agent if the grasping point is far from the target. It is worth noticing that the obtained policy makes the trained agent able to reach the target position $\mathbf{p}_F$ with 100% success rate, which means that it has correctly learned to perform the task. The fact that the tumor is not always exposed, even if the agent grasps the tissue, depends on the fact that no visual cues are considered at this stage in the reward function. In future works we plan to integrate the current reward function with an additional term assessing the amount of exposed surface.

C. Robotic Setup

The trained policies are transferred and tested on the dVRK (Fig. 1). The real and simulated setup are initially aligned with respect to the same reference frame, defined by the center of the calibration board (Fig. 3). The mean positioning error of the PSM arm is $1.7\,mm$. We have been

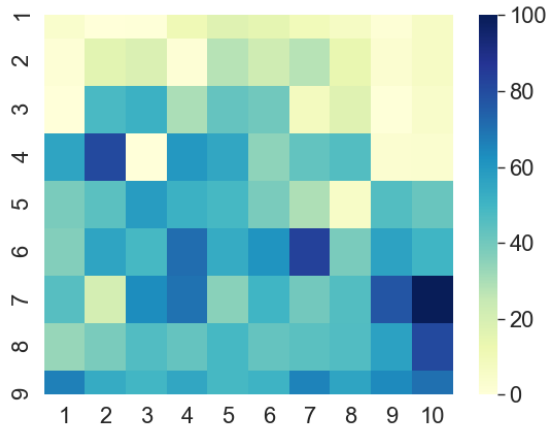


Fig. 5. Percentage of tumor area exposed to the simulated camera at different initial position of the PSM arm, uniformly distributed over the fat region. The color of each subregion is correlated with the percentage of visible tumor area when $\mathbf{p}_0$ belongs to that subregion.

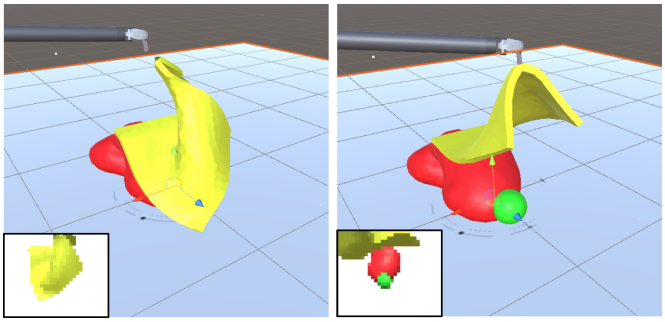


Fig. 6. Simulation results related to two different starting position of the PSM arm, with the corresponding image captured by the simulated camera. Left: EE initial position falls within subregion (2,10) of the grid in Fig. 5, and the tumor is not exposed. Right: EE initial position falls within subregion (8,6) of the grid in Fig. 5, the tumor is exposed.

able to successfully replicate the learned behavior from the simulated to the real environment without any appreciable latency. Please refer to the video attachment for further proof of the behavior. The da Vinci EE successfully gets in contact with the fat tissue for all the different initial positions, and it is always able to reach the target point. The main difference between the simulated and the real environment is related to grasping task. Figure 7 shows that the PSM can grasp the fat tissue in 9 cases out of 25, and in 2 cases the tissue is initially grasped but lost during the upward movement (semi-successful grasp). The obtained distribution suggests that the tissue is more likely to be grasped when the EE starts from positions above the tumor area. It is worth noticing that the presence of failed grasps is due to the difference between the real and the simulated grasping. In simulation, when the proximity condition is verified (Section IIIA), a grasping action is triggered and the fat is assumed rigidly attached to the EE. If the same condition could be precisely replicated in the real setup, we expect the total number of accomplished grasps to increase, due to the fact that the

PSM is able to touch the fat surface in all the attempts. In reality, grasping represents a more challenging action which should not be driven only by the relative distance between the instrument and the tissue, but also by tool orientation and tissue properties. Examples of a successful and failed grasps are shown in Fig. 8. This aspect will be addressed in future works by both improving the realism of the simulated grasping and integrating sensors in the real environment able to provide online feedback on the grasping status. Furthermore, we plan to evaluate our approach in more general settings, for example considering a variable orientation of the EE.

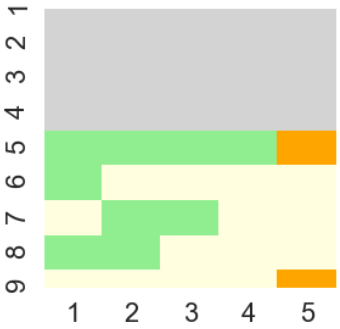
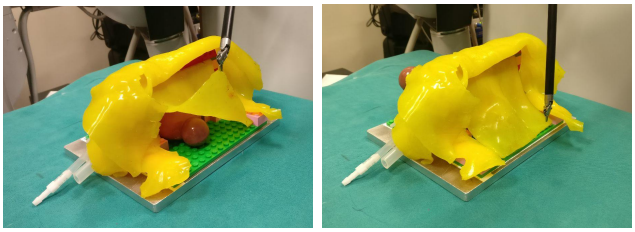


Fig. 7. Color of each cell represents the outcome of the grasping task performed with the dVRK, at different initial position of the PSM arm, uniformly distributed over the fat region. Green, orange and yellow represent a successful, semi-successful and failed grasps, respectively. The portion of fat tissue which is not considered for the experiments is colored in gray.



(a) (b)

Fig. 8. Examples of (a) successful (EE initial position in (8,1) of the grid in Fig. 7) and (b) failed (EE initial position in (9,4) of the grid in Fig. 7) grasps on the dVRK.

VII. CONCLUSIONS

In this work, we designed and implemented a flexible simulation environment suitable for DRL training in surgical robotic applications. We demonstrated the capabilities of the proposed simulation in fat tissue manipulation for the exposure of tumor during robotic assisted nephrectomy procedure. We demonstrated that a DRL agent trained in simulation is able to generalize to real scenario thanks a calibration procedure for optimizing the simulation parameters to reduce the reality gap. Results confirm the robust performance and generalization capabilities of the trained method. The proposed simulation environment is an essential component in development of autonomous agents for controlling surgical tools and manipulating soft tissues. Future work would be focused on autonomous control using visual cues, which would allow to implicitly account for the current deformed tissue configuration. We also plan to extend the work towards using model-based approaches and incorporating imitation based learning to bootstrap the number of training steps.

REFERENCES

[1] H. F. Roh, S. H. Nam, and J. M. Kim, "Robot-assisted laparoscopic surgery versus conventional laparoscopic surgery in randomized controlled trials: a systematic review and meta-analysis," *PLoS One*, vol. 13, no. 1, 2018.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3387–3395.

[4] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2371–2378.

[5] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," *arXiv preprint arXiv:1903.02090*, 2019.

[6] A. Boeing and T. Bräunl, "Leveraging multiple simulators for crossing the reality gap," in *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2012, pp. 1113–1119.

[7] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4111–4117.

[8] T. Osa, K. Harada, N. Sugita, and M. Mitsuishi, "Trajectory planning under different initial conditions for surgical task automation by learning from demonstration," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6507–6513.

[9] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1202–1209.

[10] T. D. Nagy, M. Takács, I. J. Rudas, and T. Haidegger, "Surgical subtask automation?soft tissue retraction," in *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2018, pp. 000 055–000 060.

[11] N. D. Nguyen, T. Nguyen, S. Nahavandi, A. Bhatti, and G. Guest, "Manipulating soft tissues by deep reinforcement learning for autonomous robotic surgery," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–7.

[12] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," *arXiv preprint arXiv:1806.07851*, 2018.

[13] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, "Autonomous tissue manipulation via surgical robot using learning based model predictive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3875–3881.

[14] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[15] E. Tagliabue, D. Dall'Alba, E. Magnabosco, C. Tenga, I. Peterlik, and P. Fiorini, "Position-based modeling of lesion displacement in ultrasound-guided breast biopsy." *IJCARS*, 2019.

[16] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2018.

[17] (2018) NVIDIA gameworks. Nvidia FleX. [Online]. Available: https://developer.nvidia.com/flex

[18] L. Qian, A. Deguet, and P. Kazanzides, "dVRK-XR: Mixed Reality Extension for da Vinci Research Kit," in *Hamlyn Symposium on Medical Robotics*, 2019.